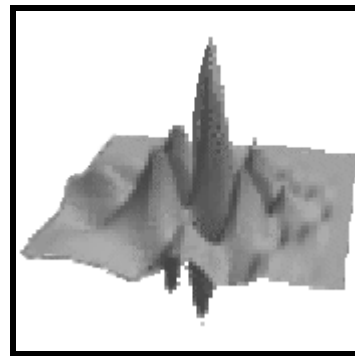


Ausführlicher Statusbericht zum Forschungsthema

**Bestimmung der Drahtvorschubgeschwindigkeit
beim MIG/MAG-Schweißen mittels
berührungsloser Messverfahren**



Technische Universität Chemnitz
Institut für
Fertigungstechnik / Schweißtechnik
Reichenhainer Straße 70
09126 Chemnitz

Tel.: (0371) 531-2440
Fax.: (0371) 531-2441

Prof. Dr.-Ing. habil. K.-J. Matthes

Bearbeiter: Dipl.-Ing. Thomas Kohler

Technische Universität Chemnitz
Fakultät für Mathematik
Arbeitsgebiet Numerik
Reichenhainer Str. 41 / 618
09107 Chemnitz

Tel.: (0371) 531-8441
Fax.: (0371) 531-2657

Prof. Dr. R. Schneider

Bearbeiter: Dipl.-Math. V. Vassiliev

DFG-Geschäftszeichen: MA 1391/17-1

Förderungszeitraum/Berichtszeitraum: 01.05.2002 – 30.04.2004

Inhaltsverzeichnis

Formelzeichen, Abkürzungen und Symbole	III
1 Einleitung.....	1
1.1 Ausgangssituation	1
1.2 Lösungsansatz	4
2 Grundlagen der Bildaufnahme	6
2.1 Der CCD-Sensor	6
2.2 Abbildungsfehler von CCD-Sensoren	6
2.3 Die Videonorm.....	7
2.4 Abtastung des Bildsignals	7
2.5 Die Bildqualität	8
2.6 Kameratypen	8
2.7 Optische Grundlagen.....	9
3 Der Versuchsstand	11
3.1 Funktionsweise.....	11
3.2 Aufbau des Versuchsstandes.....	12
3.2.1 Gesamtkonfiguration	12
3.2.2 Rechnerkonfiguration	12
3.2.3 Antrieb und Motorsteuerung.....	13
3.2.4 CCD-Kamera und Framegrabberkarte.....	14
3.2.5 Beleuchtung und Beleuchtungssteuerung.....	14
4 Versuchsdurchführung Teil 1	17
4.1 Verwendete Materialien und ihre Eigenschaften	17
4.2 Probleme bei der Aufnahme.....	18
4.2.1 Krümmung der Drahtachse	18
4.2.2 Offset des Kamerasignals	18
4.2.3 Aufnahmeverfahren/Aufnahmerichtung.....	18
4.3 Untersuchung verschiedener Lichtquellen	21
4.3.1 Glühlampenlicht/Kaltlicht	21
4.3.2 Diodenlaserlicht	21
4.3.3 Leuchtdiodenlicht	21
5 Mathematische Algorithmen zur Versuchsauswertung	24
5.1 Allgemeines zu den Aufbereitungs- und Auswertalgorithmen	24
5.1.1 Aufbereitungsalgorithmen	24
5.1.2 Auswertalgorithmen.....	24
5.1.3 Effekte durch die Nutzung von Wavelets	25
5.2 Beschreibung der mathematischen Vorgehensweise	26
5.2.1 Prinzip im Überblick.....	26

5.2.2	Notationen im Matlab	29
5.2.3	Die Qualität des Bildes	30
5.2.4	Algorithmus zur Kalibrierung.....	31
5.2.5	Algorithmus zum Deinterlacing.....	31
5.2.6	Algorithmus zur Bestimmung der Kanten des Drahtes	31
5.2.7	Algorithmus zur Berechnung der Neigung des Drahtes	33
5.2.8	Algorithmus zur Berechnung der Rotation des Bildes	33
5.2.9	Allgemeines zur Berechnung der Ähnlichkeitsfunktion.....	35
5.2.10	Zyklische Verschiebung und Korrelation	36
5.2.11	Azyklische Verschiebung	36
5.2.12	Azyklische Verschiebung mit Fenster	36
5.2.13	Azyklische Verschiebung mit Fenster und Beleuchtungskorrektur	37
5.2.14	Bemerkungen zu den Algorithmen	37
5.2.15	Beschreibung des lokalen Minimums einer diskreten Funktion.....	38
5.2.16	Schema des gesamten Algorithmus	38
5.2.17	Algorithmen zur Statistischen Kontrolle	39
5.2.18	Beschleunigung des Algorithmus mit Hilfe der Wavelet Transformation	39
5.2.19	Beschleunigung mit Hilfe der Mittelwert-Transformation in y-Richtung	40
5.2.20	Schema des Schnellen Algorithmus	40
5.2.21	Realisierung	41
6	Versuchsdurchführung Teil 2	42
6.1	Aufnahme und Auswertung verschiedener Drahtqualitäten	42
6.1.1	Allgemeine Hinweise zur praktischen Durchführung der Versuche	42
6.1.2	Hinweise zu den Werkstoffen.....	43
6.1.3	Zusammenfassende Darstellung der Versuche	44
7	Zusammenfassung und Ausblick	47
8	Literaturverzeichnis.....	48
9	Anhang	53
9.1	Schaltplan zur synchronisierten Lichtansteuerung.....	53
9.2	BS2SX-Programmcode (Motorregler)	54
9.3	Matlab – Programmcode (Hauptprozeduren).....	59
9.4	Matlab – Programmcode (Hauptbearbeitungsprozedur).....	60
9.5	Matlab – Programmcode (Berechnung der Verschiebung).....	60
9.6	Matlab – Programmcode (Verschiebung mit Neigungskorrektur).....	64
9.7	Matlab – Programmcode (Bestimmung der Kanten)	66
9.8	Matlab – Programmcode (Ähnlichkeitsfunktionen).....	66
9.9	Matlab – Programmcode (Lokale Minima).....	68
9.10	C++ – Programmcode (Rotation).....	69
9.11	Matlab – Programmcode (Qualitätsbestimmung)	73
9.12	Matlab – Programmcode (Transformationen).....	74
9.13	Matlab – Programmcode (Verschiedenes)	76

Formelzeichen, Abkürzungen und Symbole

Formelzeichen/Begriff/Symbol	Einheit	Bezeichnung
a	<i>Pixel/mm</i>	Bildauflösung, Pixeldichte
B	<i>mm</i>	Bildgröße
b	<i>mm</i>	Bildweite
C	<i>mm</i>	zulässiger Streuungskreisdurchmesser
D	-	Flächendifferenzfunktion
d	<i>mm</i>	Blendendurchmesser
f	<i>mm</i>	Brennweite
f_B	<i>Hz</i>	Bildwiederholfrequenz
G	<i>mm</i>	Bildfeld der Kamera / Gegenstandsgröße
g	<i>mm</i>	Gegenstandsweite
g_0	<i>mm</i>	Nennabstand bei Objektivfokussierung
i	-	Laufindex, Stelle im Bild
K	-	Korrelationsfunktion
k	-	Blendenzahl
m	-	Pixelvektor aus Bild 1
n	-	Pixelvektor aus Bild 2
N	-	Pixelanzahl
R_a	μm	Arithmetischer Mittenrauwert
R_z	μm	Maximale Rauheitsprofilhöhe
s	<i>Pixel</i>	Verschiebewert
sj	<i>Pixel</i>	Index für Verschiebewert $s = 0 \dots N-1$
t	<i>mm</i>	Schärfentiefe
t_B	<i>s</i>	Belichtungszeit
U	<i>Pixel</i>	Bewegungsunschärfe
v_D	<i>m/min</i>	Drahtvorschubgeschwindigkeit
x	<i>mm</i>	Abstand / Verschiebeabstand

1 Einleitung

1.1 Ausgangssituation

Da die Drahtvorschubgeschwindigkeit einen bedeutenden Einfluss auf die Ausbildung der Nahtgeometrie hat und ebenfalls mit den elektrischen Parametern in Zusammenhang steht, nimmt die Einhaltung des vorgegebenen Geschwindigkeits-Sollwertes einen besonderen Stellenwert ein. Die Drahtvorschubgeschwindigkeit ist für den Schweißer ein wesentlicher unabhängiger Führungsparameter, der mit der Abschmelzleistung gleichzusetzen ist.

Heute eingesetzte Systeme zur Drahtgeschwindigkeitsmessung beruhen im Wesentlichen auf der Messung der Drehzahl des antreibenden Motors (s. Abbildung 1.1) mit Hilfe mehr oder weniger genauer Messsysteme (Tachometer, Inkrementalgeber). Das Ergebnis wird über den Umfang der Abtriebsrollen in die Lineargeschwindigkeit des Drahtes umgerechnet. Das Hauptproblem bei dieser Art der Messung besteht im auftretenden Schlupf zwischen Abtriebsrolle und Zusatzwerkstoff. Der Betrag des Schlupfes ist von Faktoren wie Art, Stärke und Oberflächenzustand des Zusatzwerkstoffes, Rollenmaterial und -geometrie, eingestellter Andruckkraft, Reibung im Schlauchpaket usw. abhängig. Es entsteht eine Differenz zwischen gemessener Umfangsgeschwindigkeit und tatsächlicher Drahtgeschwindigkeit. Eine Alternative wäre das Anbringen des Messsystems auf der Seite der passiven Andruckrolle (s. Abbildung 1.2).

Der Schlupf und damit die Genauigkeit des Messergebnisses sind hier „nur“ noch von der Leichtgängigkeit der Rolle-Messsystem-Kombination (Reibung und Trägheit) abhängig. Messungen haben gezeigt, dass diese Systeme sehr einfach (auch extern) realisierbar sind, aber zur Bestimmung schnellerer Änderungen sowie geringer Drahtgeschwindigkeiten aufgrund ihrer relativ großen Trägheit und begrenzten Auflösung nicht geeignet sind. Der relative Fehler solch einer Anordnung lässt sich nicht unter 10 % senken. Dies hat Auswirkungen auf den gesamten Regelkreis zur Drahtvorschubgeschwindigkeitseinstellung.

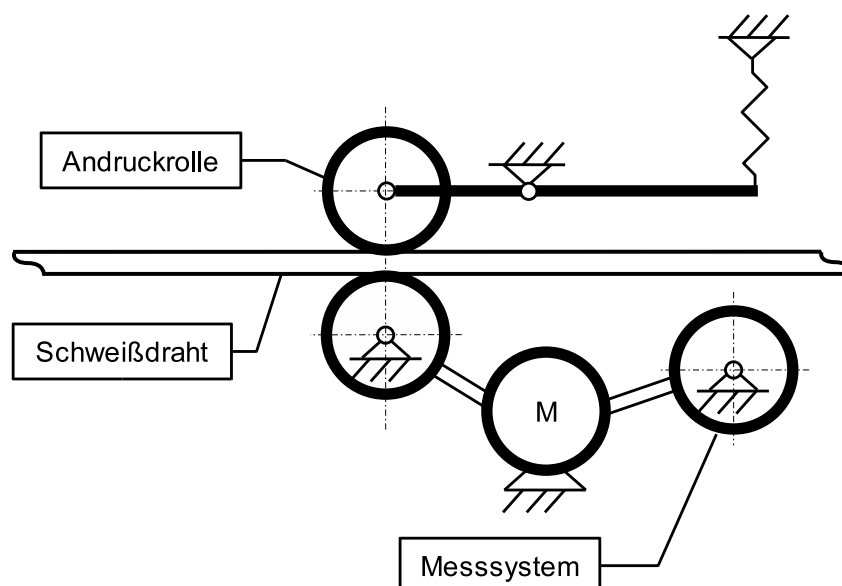


Abbildung 1.1: System zur Motorgeschwindigkeitsregelung

Bei in [6] durchgeführten Untersuchungen wurden Abweichungen der Drahtvorschubgeschwindigkeit bis 30 % ermittelt. Einen Ausweg aus dieser Situation bieten berührungslos arbeitende Messsysteme. Voraussetzung dafür sind Sensoren, welche die spezifischen Eigenschaften des Messobjektes berührungslos erfassen können bzw. dem Messobjekt ein spezielles Merkmal aufprägen (Streifenmuster o. ä.), welches durch den Sensor selbst wieder erkannt wird.

Für die erste Variante kommen, je nach Anwendungsfall, optische [7, 8], akustische (z. B. Ultraschall [9]), induktive, kapazitive, Mikrowellen- [10] und Wirbelstromsensoren [11] zum Einsatz. Optische Systeme können dabei mit Doppelbelichtung arbeiten, um den Messaufbau zu minimieren [12].

Im anderen Falle ist eine Sender-Empfänger-Kombination notwendig. Grundsätzlich sind dabei alle oben genannten Sensorprinzipien möglich, wobei spezielle Kombinationen bereits untersucht wurden bzw. auf dem Markt verfügbar sind. Das sind unter anderem akustische oder optische, nach dem Laufzeitverfahren arbeitende Sensorsysteme [13, 14]. Diese werden z. B. zur Abwasser-Rohrnetzüberwachung eingesetzt.

Da die Preise für einfache Kamerasysteme in den letzten Jahren enorm gesunken sind, die Baugrößen minimiert wurden und bei der Bildverarbeitung meist auch eine einfache visuelle Kontrolle des Rechenergebnisses möglich ist, soll für den beschriebenen Anwendungsfall eine CCD-Kamera zum Einsatz kommen (s. Abbildung 1.3). Bildverarbeitungsprozesse werden üblicherweise im Schema Bildaufnahme, Vorverarbeitung, Merkmalsextraktion und Klassifikation abgearbeitet. Im vorliegenden Falle sind die letzten beiden Schritte zu „Auswertalgorithmen“ zusammengefasst. In [15] wird darauf hingewiesen, dass insbesondere im Hinblick auf die Komplexität der Algorithmen, der Einfluss der Beleuchtung ein wichtiges Kriterium der digitalen Bildverarbeitung von Grauwertbildern darstellt. Daher wird dieser Bereich gesondert im Abschnitt 4.3 untersucht. Um die Signale der Sensoren auswerten zu können, kommen aufwändige Berechnungsalgorithmen zur Anwendung, die meist nur durch hohe finanzielle Aufwendungen (z. B. mehrere Kameras [16]) Echtzeitfähigkeit erreichen. Das in [17] vorgestellte Verfahren berechnet das Ausgangssignal auf einem Multiprozessorsystem (Transputersystem) und erreicht damit Messwiederholraten von 200 ms. Eine maximale Messdauer für das Drahtvorschubsystem zwischen 50 ms und 100 ms dürfte realistisch sein. Die herkömmlichen Berechnungsalgorithmen müssen daher beschleunigt werden. Hierfür sollen Wavelets zum Einsatz kommen.

Wavelets sind ein relativ neuartiges mathematisches Werkzeug in der Signalverarbeitung. Orthogonale und biorthogonale Wavelets mit kompaktem Träger wurden von I. Daubechies Ende der 80er Jahre eingeführt. Von praktischer Bedeutung ist die so genannte (schnelle) Wavelettransformation. Während die klassische Fouriertransformation ein Signal in seine Spektralanteile zerlegt, wird durch die Wavelettransformation das Signal in verschiedene Skalenanteile aufgespalten. Die lokale Information geht dabei nicht ganz verloren. Daher erweist sich die Wavelettransformation bei Signalen mit ausgeprägten lokalen Charakteristika, wie zum Beispiel scharfe Konturen, als besonders vorteilhaft. Dies wird zur Bildkompression, also zur Datenreduktion, wie auch zum Unterdrücken von Rauscheffekten mit Erfolg eingesetzt und ist Gegenstand intensiver aktueller Forschung. Vom mathematischen Verständnis her beruhen die Vorzüge auf der so genannten nicht-linearen Approximation [18, 19]. Hierbei wird eine möglichst optimale Approximation mit einer festen Anzahl N von Entwicklungskoeffizienten gesucht, die beste N -Term Approximation.

Dieses Konzept kann man noch konsequenter erweitern, indem man eine Bibliothek von Basisfunktionen zugrunde legt und nach einer quasi-besten Basis für ein konkret vorliegendes Signal sucht. D. h. man versucht mit möglichst wenigen Freiheitsgraden eine möglichst genaue Approximation zu erzielen. Diese Prinzipien und auch ihre Verbindung zu statistischen Schätzmethoden sind in dem Buch von Mallat [4] ausgeführt. Leider

weisen mehrdimensionale Signale, wie beispielsweise (2-dimensionale) Bilder, mit Sprüngen (d. h. scharfe Konturen) eine nicht optimale Regularität im Sinne der Besov-normen auf. Hierfür sind Wavelets im Allgemeinen immer noch von einer theoretisch optimalen Lösung entfernt, und wie in [20] bemerkt wird, besteht hier noch Handlungsbedarf. Liegen die Sprünge allerdings parallel zu den Koordinatenrichtungen, so liegt eine Regularität in anisotropen Besovräumen vor, die mit Tensor-Produkt-Wavelets effektiv behandelt werden können. Solche Techniken sind in letzter Zeit auch in der Numerik unter dem Begriff „Sparse Grids“ populär geworden. D. h. eine tensorielle zweidimensionale Wavelettransformation verspricht im vorliegenden Anwendungsfall ein noch größeres Potenzial der Kompression, denn die Bilder weisen typischerweise eine hierfür geeignete Anisotropie auf. Bei der effizienten numerischen Behandlung von Integraloperatoren mittels Matrixkompression, dem eigentlichen Spezialgebiet des Mitantragstellers, treten mitunter ähnliche Effekte auf [21]. Daher sollen Wavelets zum Einsatz kommen, um das vorliegende Problem zu lösen.

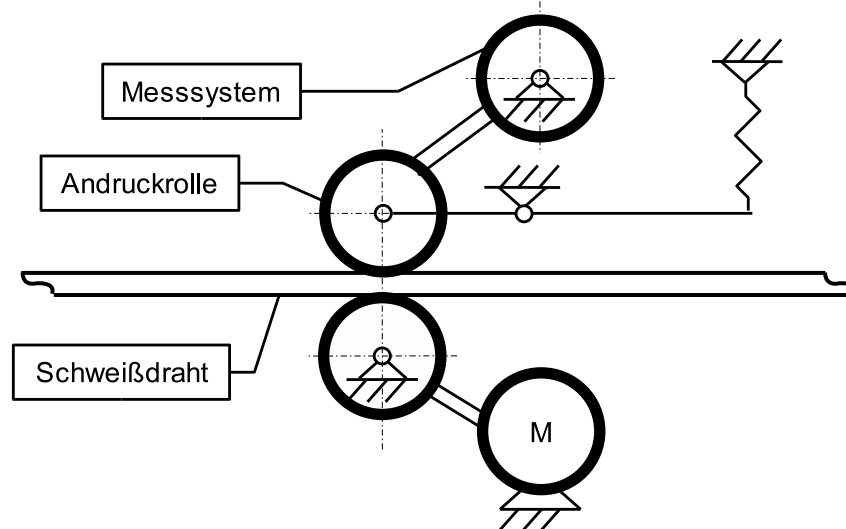


Abbildung 1.2: Drahtvorschubgeschwindigkeitsmessung über die Andruckrolle

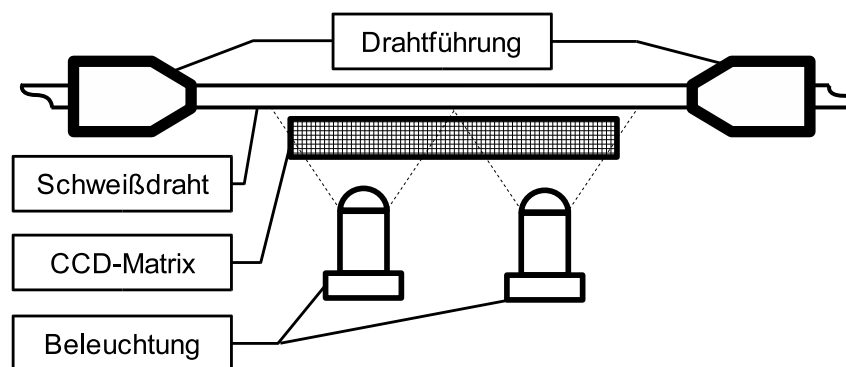


Abbildung 1.3: Berührungslose Drahtvorschubgeschwindigkeitsmessung

1.2 Lösungsansatz

An einigen Beispielbildern einer Videosequenz, welche mit Hilfe einer HI8-Kamera (470.000 Bildelemente) von Drahtoberflächen aufgezeichnet wurden lässt sich das Prinzip der Auswertung am Besten verdeutlichen. Nach Digitalisierung auf ein Format von $320 \times 256 \text{ Pixel}$ und Farbreduktion sowie Helligkeitskorrekturen mit einfachen Bildverarbeitungsprogrammen werden einige Einzelbilder auf deren Auswertbarkeit mittels Korrelationsverfahren und Flächendifferenzbildung getestet. Ziel ist es, den Verschiebvektor zweier im Zeitabstand von $1/25 \text{ s}$ aufgenommener Bilder zu ermitteln. Zur besseren visuellen Kontrolle der Rechenergebnisse wurde ein Drahtabschnitt mit fehlerhafter Oberfläche ausgesucht (s. Abbildung 1.4 und Abbildung 1.5). Zu sehen sind Bildausschnitte im Format 320×80 (nur Drahtoberfläche ohne störenden Hintergrund), welche noch manuell um zwei bis drei Grad gedreht wurden, um Parallelität zu erreichen. Damit wird eine Auflösung von $13,33 \mu\text{m} / \text{Pixel}$ erzielt. Rein visuell lässt sich eine Merkmalsverschiebung von 150 Pixeln ausmachen. Die dargestellten Bilder wurden, ohne Beachtung der Beleuchtungsverhältnisse, mit Hilfe der Korrelation und Flächendifferenzbildung ausgewertet (s. Abbildung 1.6 und Abbildung 1.7, Beschreibung im Abschnitt 5.2). Beide Verfahren liefern für diese Bilder mit eindeutigen Merkmalen das gleiche Ergebnis von 154 Pixeln Längenunterschied in $1/25 \text{ Sekunden}$. Damit lässt sich die Drahtvorschubgeschwindigkeit zu $v_D = 3,08 \text{ m/min}$ ermitteln. Im Rahmen dieser Arbeit sollen die Grundlagen und Randbedingungen beim Einsatz von CCD-Sensoren zur berührungslosen, rückwirkungsfreien, und hoch genauen Bestimmung der Drahtvorschubgeschwindigkeit ermittelt werden. Der Einsatz dieses Messsystems ist bereichsübergreifend in allen Anwendungsfällen denkbar, in denen eine berührende Messung der Objektgeschwindigkeit nicht möglich bzw. mit zu hohen Fehlern verbunden wäre. Ein Beispiel dafür ist die Messung der Geschwindigkeit von heißen Walzblechen sowie die Bestimmung der Umfangsgeschwindigkeit von schnell laufenden Rädern und Scheiben (Schleifen, Bohrungsdrücken usw.). Für einen zukünftigen Praxiseinsatz sind die Anforderungen an Aufnahme- und Auswertesystem, mechanischen Aufbau sowie Kosten zu analysieren. Weiterhin sind die Auswirkungen von Verunreinigungen zu untersuchen sowie die notwendige Auflösung zur Sicherstellung der Genauigkeit zu ermitteln.



Abbildung 1.4: Momentaufnahme 1 einer Videosequenz



Abbildung 1.5: Momentaufnahme 2 einer Videosequenz

Da das System in der Praxis bei verschiedenen Zusatzwerkstoffen und Oberflächenzuständen zuverlässig arbeiten soll, ist deren Einfluss auf das Messergebnis zu prüfen. Es sind geeignete Softwarealgorithmen zur Bildverarbeitung zu entwerfen und deren Tauglichkeit für den Praxiseinsatz abzuschätzen bzw. zu testen. In Abgrenzung zu anderen Methoden der 2D-Geschwindigkeitsmessung sind Vor- und Nachteile des beschriebenen Systems herauszuarbeiten.

Hoch genaue Drahtvorschubgeschwindigkeitsmeseinrichtungen sind für Aufzeichnungen im Rahmen der Qualitätssicherung und Fehlersuche wünschenswert. Durch Einsatz schneller Berechnungsmethoden, z. B. der Wavelettransformation, soll Echtzeitfähigkeit erreicht werden. Damit ist eine Rückführung des Messwertes im Rahmen einer Regelung sinnvoll.

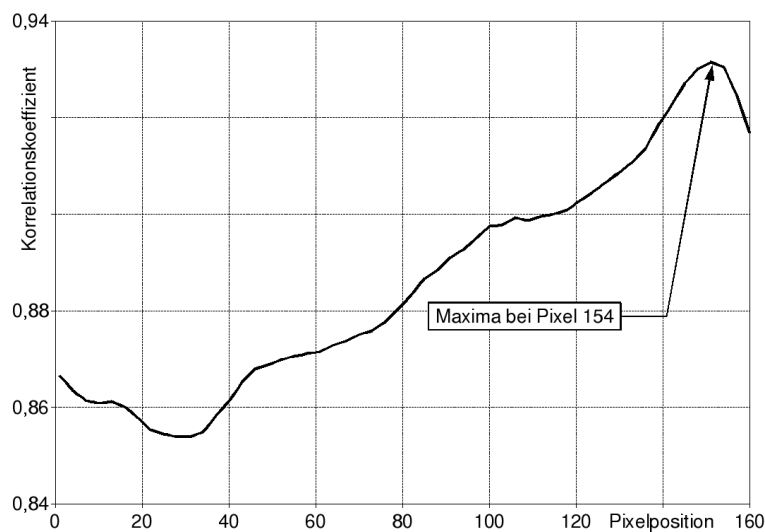


Abbildung 1.6: Ergebnis der Korrelationsberechnung

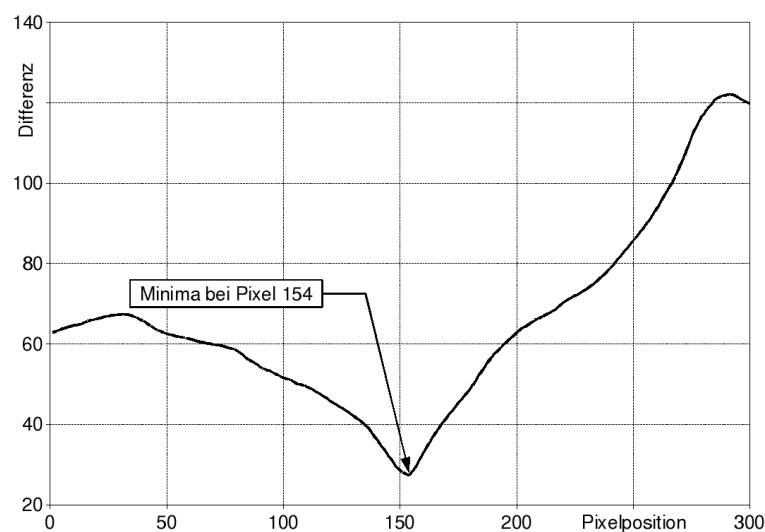


Abbildung 1.7: Ergebnis der Flächendifferenzberechnung

2 Grundlagen der Bildaufnahme

2.1 Der CCD-Sensor

Ein CCD-Sensor (charge coupled device) besteht aus vielen flächig angeordneten Halbleiterelementen (picture element oder Pixel). Die Arbeit dieser Einzelsensoren unterteilt sich in 3 Abschnitte. Sie haben die Aufgabe, einfallende Photonen in elektrische Ladung umzuwandeln, zwischenzuspeichern und aus der Sensorfläche heraus zu transportieren. Das Licht setzt Elektronen aus dem Halbleitermaterial frei. Diese werden zwischen Potenzialwällen des Sensors festgehalten (gespeichert). Die Ladungsmenge ist hierbei proportional zur Intensität des eingefallenen Lichts und zur Belichtungszeit. Die Ladungen werden aus dem lichtempfindlichen Bereich in ein Transportverzeichnis „geschoben“, wo sie nach verschiedenen Methoden ausgelesen werden können und auf das Ausgangssignal der Kamera aufmoduliert werden. Es gibt drei Typen von CCD-Sensoren.

Interline-Transfer-Sensor (IT): Dieser besitzt neben seinen Photoelementen vertikale Transportverzeichnisse (Schieberegister). Von dort werden die Ladungen in ein horizontales Schieberegister geschoben und nach der Videonorm ausgelesen. Der Nachteil dieses Sensors sind die blinden Flecken, die durch die vertikalen Register entstehen.

Frame-Transfer-Sensor (FT): Dieser besitzt keine vertikalen Schieberegister und somit auch keine Blindstellen. Hier wird das gesamte Bild durch den Sensor auf eine Speicherzone „geschoben“. Diese Speicherzone hat die gleiche Fläche wie der Sensor, liegt aber hinter diesem. Im Moment des Auslesens muss der Lichteinfall durch einen Shutter verhindert werden. Das Ausleseergebnis würde sonst verfälscht werden.

Frame-Interline-Transfer-Sensor (FIT): Dieser ist eine Mischung aus den beiden oben genannten Sensortypen. Hier gibt es sowohl vertikale Schieberegister als auch eine Speicherfläche. Ein Abdecken des Sensors während des Auslesens ist nicht nötig, es treten aber die oben genannten blinden Flecken auf dem CCD-Chip auf.

2.2 Abbildungsfehler von CCD-Sensoren

Dunkelstrom: Durch thermische Erregung freigesetzte Ladungsträger innerhalb der lichtempfindlichen Elemente eines Sensors bezeichnet man als Dunkelstrom. Dadurch entstehen bei langen Belichtungszeiten (z. B. in der Astronomie) typische Flecken auf dem Bild. Durch Kühlung der Kamera mit flüssigem Stickstoff oder mit Piezo-Elementen kann das Rauschen durch den Dunkelstrom verringert werden.

Übersprechen: Die einzelnen Elemente auf dem CCD-Sensor arbeiten nicht unabhängig voneinander, da sie auf einem durchgehenden, ebenfalls lichtempfindlichen Halbleitersubstrat aufgebracht sind. Elektronen, die zwischen den Pixeln angeregt werden, werden von umliegenden Elementen aufgenommen. Diese Elemente speichern demzufolge einen zu hohen Grauwert ab, das Bild wird verfälscht. Das kann bis zum sogenannten Blooming führen. Hier entsteht eine helle Fläche durch viele übersprochene Elemente. Besonders bei langwelligem (rotem) Licht tritt ein vermehrtes Übersprechen auf, da es tiefer in das lichtempfindliche Halbleitermaterial des CCD-Sensors eindringen kann.

2.3 Die Videonorm

Die Übertragung von Bildsignalen nach der Videonorm stellt derzeit den Stand der Technik dar. Standard-Videokameras erzeugen ein analoges Signal, dessen Spannungswerte der Helligkeit der einzelnen Bildpunkte entsprechen. Das Signal variiert nach der Europäischen CCIR -Norm zwischen 0 V (schwarz) und 0,7 V (weiß). Es werden erst alle ungeraden dann alle geraden Bildzeilen ausgelesen. Dabei ist auf dem Ausgangssignal der Anfang jedes Halbbildes (vertikale Austastlücke) und der Anfang jeder Zeile (horizontale Austastlücke) durch einen Impuls gekennzeichnet. Ein Empfänger (z. B. Fernseher oder Framegrabber) muss diese Impulse erkennen und aus den beiden Halbbildern ein Vollbild zusammensetzen. Es tritt sowohl bei der Aufnahme als auch bei der Abbildung des Videosignals immer ein zeitlicher Versatz von 20 ms zwischen den beiden Halbbildern auf. In jeder Sekunde werden 25 Vollbilder übertragen, also 50 Halbbilder. Man spricht in diesem Fall von 50 Hz-Geräten (s. Tabelle 2.1). Manche Kameras oder Framegrabber verwenden eine erweiterte Videonorm, um eine höhere Auflösung des Bildes (bei speziellen Vermessungsaufgaben evtl. notwendig) zu erreichen. Diese Erweiterungen beziehen sich auf die Abtastrate, das Zeitraster oder die Zeilenfolge (siehe Abschnitt 2.6)

Tabelle 2.1 Eigenschaften der CCIR-Norm

Kategorie	Beschreibung
Bildfolge	25 Vollbilder/s
Bildübertragung	Interlaced (50 Halbbilder/s)
Bildaufnahmezeit	Vollbild 40 ms
Seitenverhältnis	Breite : Höhe = 4:3
Zeilenzahl	625 Zeilen/Vollbild
Zeilendauer	64 μ s
Signalbandbreite	5 MHz

2.4 Abtastung des Bildsignals

Zur digitalen Bildverarbeitung muss das analoge Videosignal in ein digitales Signal umgewandelt werden. Der Abtastung ist größte Bedeutung beizumessen, da sie das entstehende Bild grundlegend beeinflusst. Bei der Abtastung mit einer Frequenz von 10 MHz werden aus einer Bildzeile 640 Werte gewonnen. Davon entfallen ca. 120 Werte auf den Zeilenanfangsimpuls, so dass eine digitale Bildzeile am Ende noch rund 520 Bildpunkte besitzt. Ändert man die Abtastfrequenz, so ändert sich auch die Zahl der Bildpunkte pro Zeile. Die Breite des Bildes ändert sich, während die Höhe aufgrund der in der Norm vorgegebenen Anzahl der Zeilen konstant bleibt. Damit ergibt sich eine Verzerrung des Bildes in Bezug auf das Verhältnis Breite zu Höhe. Zudem sind die Pixel nicht mehr quadratisch und genaue Längenmessungen an einem solchen Bild werden beeinträchtigt. Mit Hilfe von Bildverarbeitungshilfsmitteln kann zwar eine Umrechnung auf die richtigen Proportionen erfolgen, der Informationsverlust durch die fehlerhafte Abtastung jedoch ist nicht mehr kompensierbar. Bei der Abtastung treten auch Schwankungen der Abtastfrequenz auf. Die exakte Position des Bildpunktes innerhalb des

Zeitrasters ist demnach nicht bekannt. Dieser Fehler heißt Pixel-Jitter und ist nur bei der subpixelgenauen Detektierung z. B. von Körperkanten signifikant. Die richtige Wahl der Abtastrate ist wichtig für die Wiedergabegenauigkeit der entstehenden Bilder. Die Abstimmung von Kamera und Framegrabber lässt sich mit einem Oszilloskop durchführen, ist aber aufwändig und kann bei schwankenden Temperaturen erheblich abweichen. Eine weitere Möglichkeit ist, den Auslesetakt der Kamera mit dem des Framegrabbers zu koppeln. Dazu muss die Kamera ihren Takt ausgeben und der Framegrabber diesen Takt einlesen können. Es gibt heutzutage eine Reihe von Geräten, die diese pixelsynchrone Abtastung zulassen und somit ein korrektes Abbildungsverhältnis gewährleisten.

2.5 Die Bildqualität

Die Bildqualität wird neben der Qualität der verwendeten optischen Elemente hauptsächlich von 2 Kriterien, der Helligkeitsdynamik und der Ortsauflösung, geprägt.

- Die Helligkeitsdynamik ist in der Videonorm vorgeschrieben, sie kann aber durch Leitungseinflüsse beeinträchtigt werden. Jedes Übertragungselement, vom Videokabel über zwischengeschaltete Monitore bis zum Framegrabber, stellt einen Tiefpass-Filter dar, welcher die Signalfanken abrundet. Dadurch entstehen Schärfeverlust, und Synchronisationsfehler.
- Die Ortsauflösung des Bildes ist durch die Pixelzahl begrenzt. Das Bild kann nicht mehr Bildpunkte enthalten als es Sensorelemente auf dem CCD-Chip der Kamera gibt.

2.6 Kamertypen

Alle bisherigen Angaben gelten für die normale Interlaced-Video-Kamera. Es gibt jedoch noch andere Kamertypen, die sich hauptsächlich in der Übertragungsart des Bildsignales und der Geometrie des CCD-Sensors unterscheiden. Alle Kameras können auch als intelligente Kameras gebaut werden, dabei besitzen sie einen internen Rechner, der die gesamte Bildverarbeitung oder Teile davon bereits im Gehäuse der Kamera übernimmt. Sollte es nötig werden, den Versuchsstand mit einer anderen Kamera zu betreiben, dann könnten nachfolgend beschriebene Modelle eingesetzt werden. Sie sind jedoch ausnahmslos teurer als die Standard-Videokamera und die Frage nach der Wirtschaftlichkeit sollte vorerst geklärt werden.

Progressive-Scan-Kamera: Hier wird das Vollbild nicht in 2 Halbbilder unterteilt. Es wird so ausgelesen, wie es auf dem Sensor abgebildet ist, also Zeile für Zeile. Dadurch entsteht kein zeitlicher Versatz zwischen den Zeilen der Aufnahme. Diese Kameras eignen sich zur Aufnahme und Analyse schnell bewegter Vorgänge.

Asynchronkameras: Anstatt einen kontinuierlichen Signalstrom, wie bei normalen Kameras üblich, zu erzeugen, kann die Asynchronkamera zu einem bestimmten Zeitpunkt ausgelöst werden. Der Sensor kann innerhalb von 1 ms belichtet werden, worauf das Bild dann an den Framegrabber übertragen werden muss. Das wiederum dauert entsprechend der Videonorm 40 ms. Bei einem kontinuierlichen Datenstrom muss immer erst abgewartet werden bis ein Bild vollständig übertragen worden ist, bevor man ein neues anfordern kann. Das führt zu Verzögerungen. Die Asynchronkamera und ihr Framegrabber hingegen werden extern getriggert und geben sofort das Bild aus. Sie wird daher bei schnellen Vorgängen eingesetzt und ist meistens als Progressive-Scan-Kamera ausgeführt.

Digitalkamera: Für sehr genaue Messungen ist die Digitalkamera geeignet. Sie wandelt bereits das analoge CCD-Sensorsignal in ein digitales Signal um und übergibt dieses an den Framegrabber. Es können demnach keine Leitungsverluste und Bildverzerrungen mehr auftreten. Allerdings ist ein erhöhter Steueraufwand notwendig. Die Kamera muss ihr Bildsignal, ihren Pixeltakt und ihre Synchronisationssignale übertragen. Daher sind diese Systeme sehr teuer.

2.7 Optische Grundlagen

Auf dem CCD-Sensor muss ein scharfes, kontrastreiches Bild der zu prüfenden Bildszene entstehen. Um dies zu erreichen benutzt man geeignete Objektive. Zur Berechnung der Bild- und Gegenstandsweiten sowie deren Größen können, unter Annahme des Vorhandenseins dünner Linsen, die Gleichungen der optischen Geometrie verwendet werden. Jedoch sind an realen Objektiven alle Abstände auf die Hauptebenen der Linsen zu beziehen. Die Auswahl des richtigen Objektivs ist von einer Reihe von Faktoren abhängig, z. B. der Objektgröße, dem einzuhaltenden Abstand, den Beleuchtungsbedingungen, der Bauform der Kamera und besonderen Prüfungsanforderungen. Im Folgenden werden einige Begriffe erklärt, die für die Arbeit mit Bildverarbeitungssystemen und deren Optimierung nötig sind.

Blendenzahl: Verhältnis zwischen Durchmesser der Blende d und Brennweite f des Objektivs. Die Blendenzahlen k sind in der internationalen Blendenzahlreihe genormt. Von Blendenzahl zu Blendenzahl halbiert sich jeweils die einfallende Lichtmenge.

$$k = \frac{f}{d}; k \in (1;1,4;2;2,8;4;5,6;8;11;16;22;32) \quad (2.1)$$

Lichtstärke: Die größte relative Öffnung eines Objektivs heißt Lichtstärke. Sie ist Ausdruck für die minimale Beleuchtung, bei der Aufnahmen mit diesem Objektiv noch möglich sind.

Abbildungsmaßstab: Verhältnis der Bildgröße B zur Gegenstandsgröße G . Der Abbildungsmaßstab entspricht dem Verhältnis von Bildweite b zur Gegenstandsweite g .

$$\frac{B}{G} = \frac{b}{g} \quad (2.2)$$

Abbildungsgleichung für dünne Linsen:

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b} \quad (2.3)$$

Bildgröße: Diese berechnet sich wie folgt aus Gleichung (2.2) und Gleichung (2.3).

$$B = \frac{f}{f - g} \cdot G \quad (2.4)$$

Bildfeld: Das Bildfeld der Kamera gibt an, welche Gegenstandsgröße bei bestimmter Brennweite und Gegenstandsweite noch auf dem Sensor abgebildet werden kann. Dazu sind die Abmaße des Sensors aus dem Datenblatt der Kamera zu entnehmen. Setzt man Gleichung (2.2) in Gleichung (2.3) ein und stellt nach G um so erhält man den Wert des Bildfeldes.

$$G = \frac{g - f}{f} \cdot B \quad (2.5)$$

Schärfentiefe: Laut der Abbildungsgleichung (2.3) ist eine mathematische Schärfe nur in einer bestimmten Bildweite gegeben. Ist der CCD-Sensor nicht genau in dieser Bildebene angeordnet, schneiden sich die Lichtstrahlen vor oder hinter der Sensorebene. Es hat sich aber herausgestellt, dass es einen Bereich um die exakte Bildweite gibt, in dem eine ausreichende Schärfe des Bildes vorhanden ist, obwohl ein bspw. punktförmiges Bild als Zerstreuungskreis dargestellt wird. Solange dieser Kreis nicht größer ist als ein Pixel auf dem Sensor, wird er weiterhin als Punkt abgebildet. Der Bereich, um den man die Bildweite verändern kann, aber trotzdem ein Punkt auch als Punkt aufgenommen wird, heißt Schärfentiefe t und kann nach Gleichung (2.6) ermittelt werden.

$$t \approx 2 \cdot C \cdot k \cdot g_0 \cdot \frac{g_0 - f}{f^2} \quad (2.6)$$

Aufnahmesituationen: Nach [26] gibt es 3 Aufnahmesituationen, die verschiedenen Ansprüchen an die Bildverarbeitung gerecht werden:

- Fernaufnahmen: $g \gg f; b \approx f$
- Makroaufnahmen: $g \approx f; b \approx 2f$
- Mikroaufnahmen: $g \approx f; b \gg f$

3 Der Versuchsstand

3.1 Funktionsweise

Der Versuchsstand soll aufzeigen, dass eine Möglichkeit der Ermittlung der Geschwindigkeit des Schweißdrahtes beim MSG-Schweißen mit Hilfe eines berührungslosen Messsystems besteht. Alle für die Geschwindigkeitsbestimmung benötigten Daten werden von einer Videokamera erzeugt. Dabei wird der Draht stark vergrößert aufgenommen und auf Unregelmäßigkeiten (z. B. Walzstruktur) überprüft. Beim Betrieb des Versuchsstandes werden ausschließlich Mikroaufnahmen aufgenommen und ausgewertet (vgl. dazu Abschnitt 2.7).

Kamerabedingt vergeht eine konstante Zeit zwischen der Aufnahme zweier Bilder (40 ms bei Kameras nach der Videonorm). Dadurch wird die selbe Unregelmäßigkeit auf mehreren Bildern mit Zeitversatz aufgenommen. Die Bilder werden nachbearbeitet und die Position von Merkmalen auf mindestens zwei Bildern ermittelt. Dann wird der Abstand x zwischen einem Merkmal auf dem ersten und auf dem nächsten Bild gemessen (zur Bestimmung von x vgl. Abschnitt 5.2).

Die Geschwindigkeit des Objektes ist der Quotient aus dem Abstand und der verstrichenen Zeit zwischen den Bildern. Dabei hängt die Zeit von der Auswertemethode ab. Wertet man die Vollbilder aus, so ergibt sich:

$$v_D = \frac{x}{40ms} \cdot 60s / \min \quad (3.1)$$

Wertet man hingegen die Halbbilder aus, berechnet sich die Geschwindigkeit zu:

$$v_D = \frac{x}{20ms} \cdot 60s / \min \quad (3.2)$$

Der Nachteil des Versuchsprinzips ist die Begrenzung der Objektgeschwindigkeit. Sobald das Merkmal nur noch auf einem Bild vorhanden ist, ist eine Ermittlung der Geschwindigkeit nicht mehr möglich. Jedoch ist aus Gleichung (3.1) und Gleichung (3.2) zu erkennen, dass die maximal auswertbare Geschwindigkeit bei der Halbbildanalyse verdoppelt werden kann.

$$v_{\max VB} = \frac{G}{40ms} \cdot 60s / \min \quad (3.3)$$

Aus diesem Grunde sollte die Halbbildauswertung angestrebt werden. Die maximale Geschwindigkeit ist abhängig vom Bildfeld G der Kamera. Die Berechnung der Höchstgeschwindigkeit erfolgt nach:

$$v_{\max HB} = \frac{G}{20ms} \cdot 60s / \min \quad (3.4)$$

3.2 Aufbau des Versuchsstandes

3.2.1 Gesamtkonfiguration

Der Versuchsstand besteht aus 7 Baugruppen, dem Antriebsmotor, dessen Steuereinheit, dem Kameraraum, der Beleuchtungseinrichtung, deren Steuerschaltung und der Kamera. Zudem wird noch ein PC zum Verändern der Antriebseigenschaften des Motors sowie zur Aufnahme, Verarbeitung und Speicherung der Bilder benutzt (s. Abbildung 3.1).

Als Antrieb wird eine herkömmliche Drahtvorschubeinheit, bestehend aus einem Elektromotor und einem Getriebe verwendet, die über die Steuereinheit zu beliebigen Aus- und Einfahrbewegungen angeregt werden kann. Über den Messrechner hat man die Möglichkeit, die Geschwindigkeit und die Wegstrecke beliebig einzustellen. Zusätzlich können künstlich Störgrößen aufgebracht werden, wie zum Beispiel ein sinus- oder trapezförmiger Geschwindigkeitsverlauf, der zu einer konstanten Grundgeschwindigkeit addiert wird. Auch eine Drehrichtungsumkehr ist möglich. Es können also beliebige Extremsituationen sowie der normale Betrieb simuliert werden.

Die meisten Bauteile des Versuchsstandes sind aus Aluminium gefertigt. Eine Ausnahme bildet die Deckplatte des Kameraraumes, welche aus Hartgewebe besteht. Das ist notwendig, weil die Kamera nicht elektrisch leitend mit dem Motor verbunden sein darf. Bei Betrieb des Motors erscheinen im Falle der elektrischen Leitung Störstreifen auf den aufgenommenen Bildern.

3.2.2 Rechnerkonfiguration

Die vorhandene Rechnerkonfiguration ist für die Bewertung der Bildqualität und der erreichten Echtzeitfähigkeit von entscheidender Bedeutung. Die hier vorliegende Zusammenstellung der Rechnerkomponenten ist nachfolgend kurz dargestellt (s. Tabelle 3.1 und Tabelle 3.2).

Tabelle 3.1: Verwendete Rechner-Hardwarekonfiguration

Mainboard	DFI	K6XV3+/66 Rev. 3+
Chipsatz	VIA	MVP3 AGP
CPU	AMD	K6-2 500 MHz
RAM	SDRAM	256 MB
Framegrabber	FALCON	Bt848 - Chip

Tabelle 3.2: Verwendete Rechner-Softwarekonfiguration

Betriebssystem	Linux	Red Hat 8.0; Kernel 2.4.19
Motorsteuerung	Parallax	Stampw V1.096
Video-Capture	xawtv-3.74-4	streamer
Matheprogramm	MathWorks	Matlab V12.1

3.2.3 Antrieb und Motorsteuerung

Zum Einsatz kam ein Maxon-Motor an dem das Getriebe und die eigentlichen Antriebsrollen als Kompakteinheit angeflanscht sind (RA Drive, s. Abbildung 3.2). Dies gewährleistet kurze, steife Übertragungswege und somit minimale Störeinflüsse der Antriebseinheit an sich. Die Hardwarekonfiguration des Versuchsstandes ist in Tabelle 3.3 wiedergegeben.

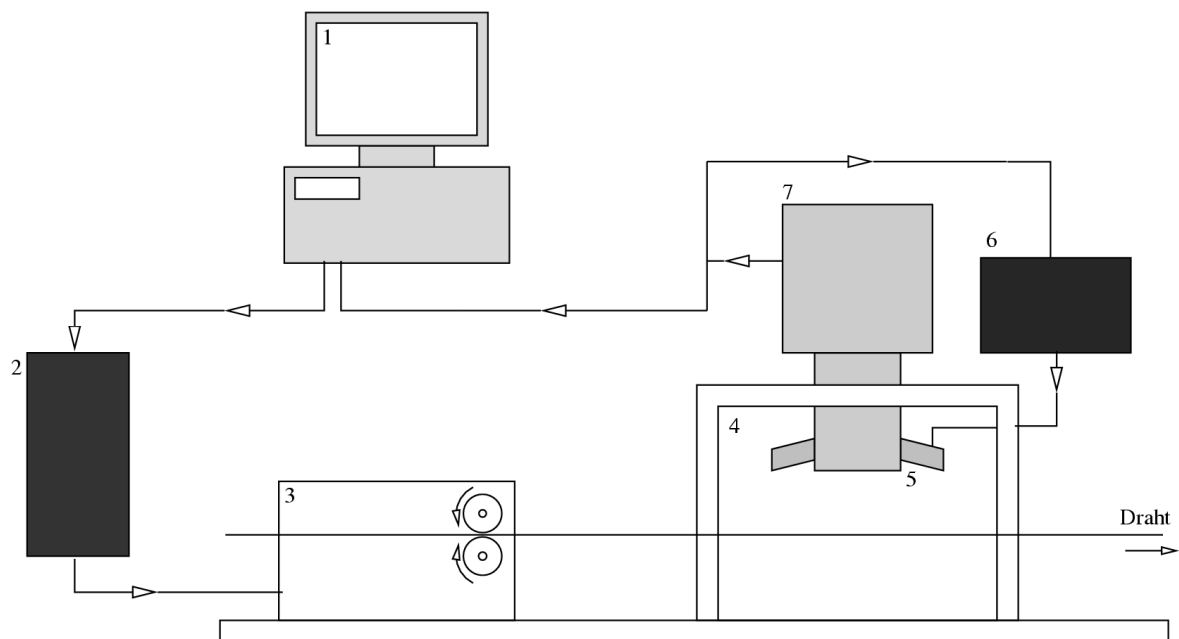


Abbildung 3.1: Aufbau des Versuchsstandes mit 1) Messrechner, 2) Motorsteuerung, 3) Motor mit Getriebe, 4) Kameraraum, 5) Beleuchtungseinrichtung, 6) Beleuchtungssteuerung, 7) CCD-Kamera

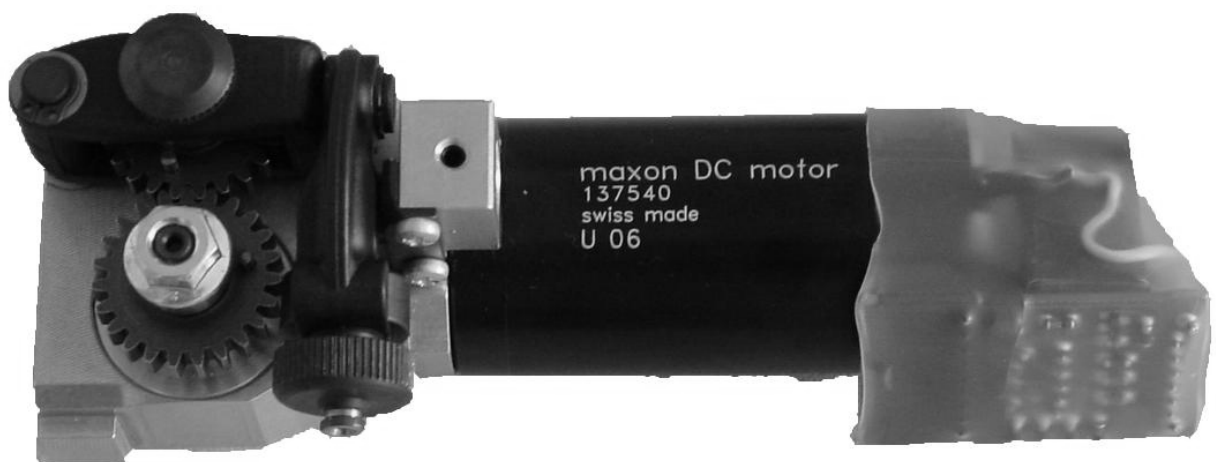


Abbildung 3.2: Am Versuchsstand eingesetzte Antriebseinheit (hier ohne Antriebsrollen)

Tabelle 3.3: Verwendete Versuchsstand-Hardwarekonfiguration

Vorschubeinheit	Fronius	RA Drive
Servoverstärker	ESR	BN6508
Motordrossel	ESR	BN3750, 2,5 mH
Steuerung	Parallax	Mikrocontroller BS2SX

Software

Die Software wurde in Basic geschrieben (s. Anhang S. 54). Darin sind das Lesen des angeflanschten Inkrementalgebers, ein PID-Regler für die Geschwindigkeitsregelung sowie ein Sollwertgenerator realisiert. Es lassen sich damit konstante Sollwerte realisieren, welche zusätzlich durch eine Dreieck-, Sinus- oder Rechteckfunktion überlagert werden können. Über eine serielle Schnittstelle am PC lässt sich das Programm in den BS2SX-Chip einspielen.

3.2.4 CCD-Kamera und Framegrabberkarte

Am Versuchsstand wird eine Standard-Videokamera verwendet. Dabei handelt es sich um das Technisch Industrielle Miniaturmikroskop (T.I.M.M.) der Firma SPI Robotics (www.spi-robotics.de). Eine Besonderheit bei dem verwendeten Modell ist das Objektiv. Es ist kein Objektiv im herkömmlichen Sinne, welches zur Einstellung der Schärfe den Abstand der Linse zum Sensor verändert, sondern um eine Linsenordnung, die durch eine Veränderung der Brennweite ein scharfes Bild auf dem CCD-Chip erzeugt. Der Nachteil dieser Optik ist, dass der genaue Wert des Abbildungsmaßstabes nicht bestimmt werden kann. Laut Hersteller können Vergrößerungen bis zu 400:1 erreicht werden. Damit ist es möglich, die genannten Unregelmäßigkeiten auf der Drahtoberfläche aufzunehmen und auszuwerten. Die Kamera arbeitet nach dem Prinzip des *interline transfer*. Dabei wird der CCD-Sensor während des Auslesezeitraumes nicht vom Lichteinfall abgeschirmt. Er kann also komplette 20 ms pro Halbbildzyklus Photonen aufnehmen. Die Kamera sendet alle 20 ms ein Halbbild an den Framegrabber. Dort werden diese zu Vollbildern zusammengesetzt und an den Messrechner zum Speichern und zur weiteren Verarbeitungen weitergegeben. Der Draht kann das Kamerasichtfeld sowohl parallel als auch senkrecht zu den Bildzeilen durchlaufen, also in horizontaler oder vertikaler Richtung.

3.2.5 Beleuchtung und Beleuchtungssteuerung

Die optimale Position der Lichtquelle ist an der Oberseite des Aufnahmebereiches. Der Draht wird also von oben angestrahlt (Auflichtanordnung). Eine Durchlichtanordnung schließt sich von vornherein aus, da die Kamera dann nur den Schatten des Drahtes aufnehmen kann und für eine Streulichtanordnung sind die verwendeten Lichtquellen zu schwach. Selbst durch das Aufräumen der Innenwände des Kameraraumes mittels Kugelstrahlen, war keine Verbesserung zu erzielen. Problematisch bei der Beleuchtung mit Auflicht ist die gekrümmte Oberfläche des Drahtes. Es treten immer hellere Regionen (vollständige Reflexion in das Kameraobjektiv) und dunklere bis schwarze Regionen (durch unvollständige oder keine Reflexion in das Kameraobjektiv) auf. Dadurch wird die Fläche, auf der die Unregelmäßigkeiten zu detektieren sind, verkleinert.

Nach ersten Versuchen stellte sich heraus, dass bei statischer Beleuchtung der Drahtoberfläche keine Messung durchführbar ist. Die Merkmale, die es zu detektieren gilt, werden, während sie sich bewegen, ständig belichtet. Dadurch sind alle Merkmale nur unscharf zu erkennen (Abbildung 3.3). In Tabelle 3.4 ist die theoretisch errechnete Unschärfe in Pixeln bei verschiedenen Belichtungszeiten t_B und Geschwindigkeiten v_D angegeben. Die Werte wurden dabei folgendermaßen ermittelt, wobei a der Bildauflösung entspricht:

$$U = v_D \cdot t_B \cdot a \quad (3.5)$$

Die Belichtungszeit muss verkürzt werden, um eine Detektion der Geschwindigkeit zu gewährleisten. Dazu könnte einerseits ein Shutter benutzt werden oder eine Blitzlichtsteuerung. Da der Bau eines Shutters recht aufwändig ist, fiel die Entscheidung auf die Blitzlichtsteuerung. Eine elektrische Schaltung soll synchron zum Kameraktakt einen Triggerimpuls an die Lichtquelle geben, welche dann für die Länge des ausgegebenen Impulses aufleuchtet und danach wieder erlischt. Die vertikale Austastlücke auf dem Kamerasignal kann für die Triggierung des Lichts verwendet werden.

Tabelle 3.4: Bewegungsunschärfe U in *Pixeln* bei verschiedenen Geschwindigkeiten und Belichtungszeiten, $a = 154 \text{ Pixel/mm}$

Belichtungszeit [ms]	Geschwindigkeit [m/min]			
	1	5	10	20
0,05	0,13	0,64	1,28	2,57
0,1	0,26	1,28	2,57	5,13
0,5	1,28	6,42	12,83	25,67
1	2,57	12,83	25,67	52,33
2	5,13	25,67	51,33	102,67



Abbildung 3.3: Verschmieren des Bildes bei Drahtbewegung und statischer Beleuchtung

Das Videosignal der Kamera wird parallel auf den Framegrabber im Messrechner und die Lichtsteuerung (Schaltplan im Anhang 9.1) gegeben. Die Schaltung detektiert die vertikale Austastlücke jedes Halbbildes und gibt mit zeitlichem Versatz den Impuls für das Aufleuchten der Lichtquelle aus (s. Abbildung 3.4). Mit einem Potentiometer ist die Länge der Beleuchtungsdauer zwischen $0,05\text{ ms}$ und $0,1\text{ ms}$ einstellbar. Auf Grund der kurzen Belichtungszeiten und der gleichzeitigen Bewegung des Drahtes, bestehen große Differenzen zwischen den einzelnen Halbbildern (Interlaced) der Kamera. In einem Vollbild ist jedes Merkmal daher zweimal zu sehen, weil es innerhalb der 40 ms zu zwei verschiedenen Zeitpunkten belichtet wurde. Es ist daher zwingend notwendig die Halbbilder zu trennen. Danach ist es möglich, die Auswertung der Geschwindigkeit anhand eines Vollbildes (Vergleich beider Halbbilder) durchzuführen. Es könnten aber auch jeweils die geraden oder jeweils die ungeraden Halbbilder zweier aufeinander folgender Vollbilder miteinander verglichen werden um die Geschwindigkeit zu ermitteln. Mit der Beleuchtungssteuerung ist es möglich, alle oben genannten Methoden anzuwenden. Die Beleuchtungssteuerung ist zudem variabel. Es können ganz verschiedene Lichtquellen an die Leistungsseite der Schaltung angeschlossen werden. Je nach benötigter Ausgangsspannung ist ein Vorwiderstand hinzuzufügen. Ein Impuls von $0,05\text{ ms}$ Länge entspricht einer Frequenz von 10 kHz . Die Impulse treten 50 mal pro Sekunde auf. Es ist unbedingt darauf zu achten, dass die angeschlossene Lichtquelle bei diesen Frequenzen und Einschaltzeiten arbeiten kann (siehe Kapitel 4.3).

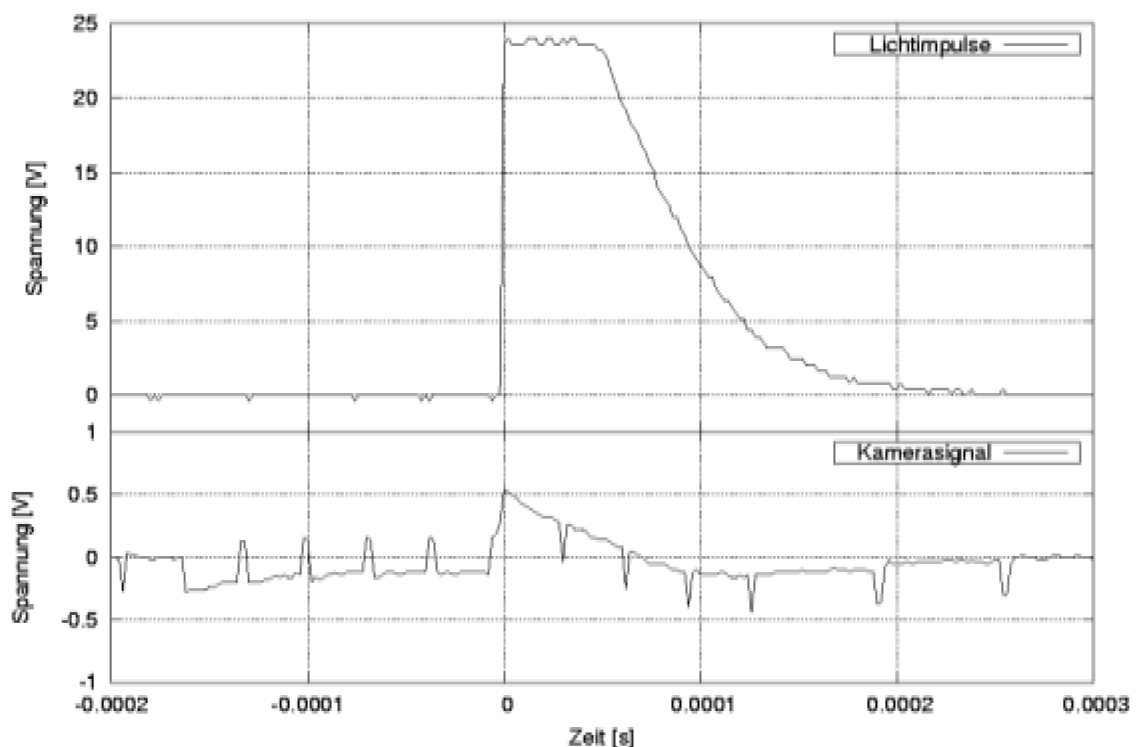


Abbildung 3.4: Durch die Lichtsteuerung detektierte Austastlücke (unten) und ausgegebener Lichtimpuls (oben)

4 Versuchsdurchführung Teil 1

4.1 Verwendete Materialien und ihre Eigenschaften

Derzeit werden zum Schutzgasschweißen am häufigsten Drahtelektroden aus unlegiertem, verkupferten Stahl und aus Aluminiumlegierungen eingesetzt. Diese Drahtelektroden sollen untersucht werden. Darüber hinaus werden ein Sonderwerkstoff, Aluminiumbronzen und andere Kupferlegierungen bei den Untersuchungen zum Einsatz kommen. Auf Grund des völlig anderen Aussehens von Fülldrähten sollen auch diese getestet werden. Tabelle 4.1 zeigt die verwendeten Drähte mit ihren Bezeichnungen.

Im Folgenden werden zur Bezeichnung der Drähte Teile aus den Kurzbezeichnungen verwendet. Ist diese sehr lang, wird die Herstellerbezeichnung genutzt. In Tabelle 4.2 sind einige Eigenschaften der verwendeten Materialien aufgeführt. Die Farbe wurde nach dem HSV-Modell angegeben, welches den Vergleich der Farbe (Hue-Wert) besser zulässt als das RGB-Modell, bei dem sich die Endfarbe aus den einzelnen Farbtönen (Rot, Grün und Blau) wie folgt zusammensetzt. Über die additive Farbmischung werden durch die 3 Grundfarben Rot, Grün und Blau alle Farben erzeugt. Im RGB Modell werden Ihre Werte je von 0 bis 1 festgelegt. $R = G = B = 1$ ergibt Weiß.

Tabelle 4.1: Verwendete Materialien und ihre Bezeichnung

Gruppe	Hersteller-bezeichnung	Werkstoff-nummer	Kurzzeichen	Hersteller
Aluminium	R 5183 (DE63)	3.3548	SG-AlMg4,5Mn	Drahtwerke Elisental
Aluminium	R 4043 A (DE59)	3.2245	SG-AlSi5	Drahtwerke Elisental
Aluminium	R 1450 (DE53)	3.0805	SG-Al99,5Ti	Drahtwerke Elisental
Stahl, verkupfert	OK Autrod 12.51	1.5125	SG-2 M2Y42 55	ESAB
Sonderwerkstoff	Nicrofer S 6025	2.4649	SG-NiCr25 FeAlY	Krupp VDM
Aluminium-bronze	AlBz8	2.0921	SG-CuAl8	Bedra
niedrig legierte Kupferlegierung	CuSi3Mn	2.1461	SG-CuSi3	Bedra
„-“ als nahtloser Fülldraht	CuSi3MnAl	2.1461	SG-CuSi3MnAl	Bedra
hoch legierter Stahl	CN22/9 N-IG	1.4462	SG-X2 CrNiMoN 22 9	Böhler
Fülldraht mit Längsnaht	EAS 4 M-FD	1.4430	T 19 12 3 L R M (C) 3	Böhler

$R = G = B = 0$ ergibt Schwarz. Grau erhält man, wenn alle 3 Werte gleich sind und zwischen 0 und 1 liegen. Monitore und Fernseher arbeiten nach diesem Prinzip. Mit einem Würfel kann man alle diese Werte darstellen (s. Abbildung 4.1). Die 3 Primärfarben und die 3 Sekundärfarben, sowie Schwarz und Weiß liegen an den Ecken (rot - grün - blau - cyan - magenta - gelb - schwarz - weiß). Das Problem am RGB-Farbmodell ist sich damit Farben vorzustellen, deshalb wurde unter anderem das HSV-Modelle entwickelt. Beim HSV-Modell (Alvy Ray Smith, 1978) stellt der Helligkeitswert V die vertikale Achse im Bereich von 0 bis 1 dar. Der Farbton H ist ein Winkel ($0...360^\circ$) um die vertikale Achse. Die Sättigung S ist der Abstand von der vertikalen Achse. Somit ergibt sich ein Zylinder mit schwarzer Grundfläche auf dessen Deckfläche alle Farben liegen (s. Abbildung 4.2).

4.2 Probleme bei der Aufnahme

4.2.1 Krümmung der Drahtachse

Ein großes Problem stellt die Krümmung der Drahtachse an sich dar. Der Draht wird beim Aufwickeln auf Rollen plastisch verformt und beschreibt daher abgewickelt immer einen Kreisbogen. Außerdem wird zusätzlich beim Einfädeln in den Versuchsstand oder durch die Antriebsrollen während des Betriebes eine Verformung eingebracht. Die Stabachse des Drahtes ist also keine Gerade. Durch die genannten Verformungen kann es vorkommen, dass sich der Draht stellenweise aus dem Sichtfeld der Kamera herausbewegt oder zu nah an das Objektiv heran kommt und dadurch unscharf abgebildet wird. Aus diesem Grunde war es unumgänglich, den Draht auf seinem Weg durch den Kameraraum zu fixieren. Dazu wurde ein Rohr mit einem Stück Teflonseele ausgebettet und in der Mitte auf einer Länge von ca. 6 mm geöffnet. Jetzt kann sich der Draht nur noch innerhalb der Seele seitlich bewegen und die Kamera nimmt durch die Öffnung die Bilder auf.

4.2.2 Offset des Kamerasignals

Je nach aufgenommener Helligkeit der Bilder wird das Kamerasignal des T.I.M.M. mit einem Offset beaufschlagt. Dadurch kann die Detektorschaltung evtl. nicht die vertikale Austastlücke erkennen und damit auch keinen Lichtimpuls auslösen. Besonders bei hohen Beleuchtungszeiten und stark verformtem Draht tritt dieser Effekt auf. Durch das Potentiometer R6 kann der Offset auf der Schaltung jedoch so kompensiert werden, dass eine einwandfreie Beleuchtung gewährleistet werden kann. Dieser Effekt tritt möglicherweise bei anderen Kameras nicht auf. Eine Testreihe, die das belegt, konnte nicht durchgeführt werden. Sollte eine andere Kamera in dem Versuchsstand Verwendung finden so ist darauf zu achten, dass eventuell der Lichtimpuls der Lichtsteuerung zum Zeitpunkt des geschlossenen Shutters aufleuchtet (bei Frame-Transfer-Kameras). In diesem Falle muss noch ein Delay zwischen vertikaler Austastlücke und Lichtimpuls auf der Schaltung eingebaut werden.

4.2.3 Aufnahmeverfahren/Aufnahmerichtung

Bei der Bildaufnahme nach dem Halbbildanalyseverfahren empfiehlt es sich, große Merkmale über mindestens 3 Vollbildzeilen zur Detektion heranzuziehen. Benutzt man kleinere Merkmale, so kann es vorkommen, dass ein Merkmal auf dem einen Halbbild abgebildet ist und auf dem nächsten nicht. Somit kann keine Geschwindigkeit errechnet werden. Das Problem tritt nicht auf, wenn der Draht sich senkrecht zu den Kameralinien bewegt. Allerdings wird der aufgenommene Drahtausschnitt verkürzt und die maximal detektierbare Geschwindigkeit verringert. Darüber hinaus ist keine Halbbildauswertung möglich.

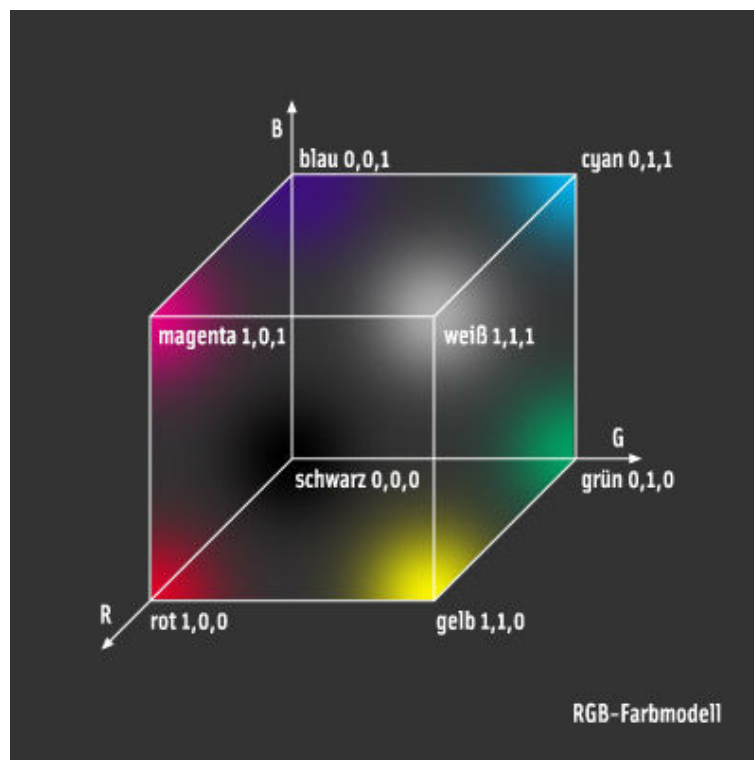


Abbildung 4.1: Das RGB-Farbmodell als Einheitswürfel

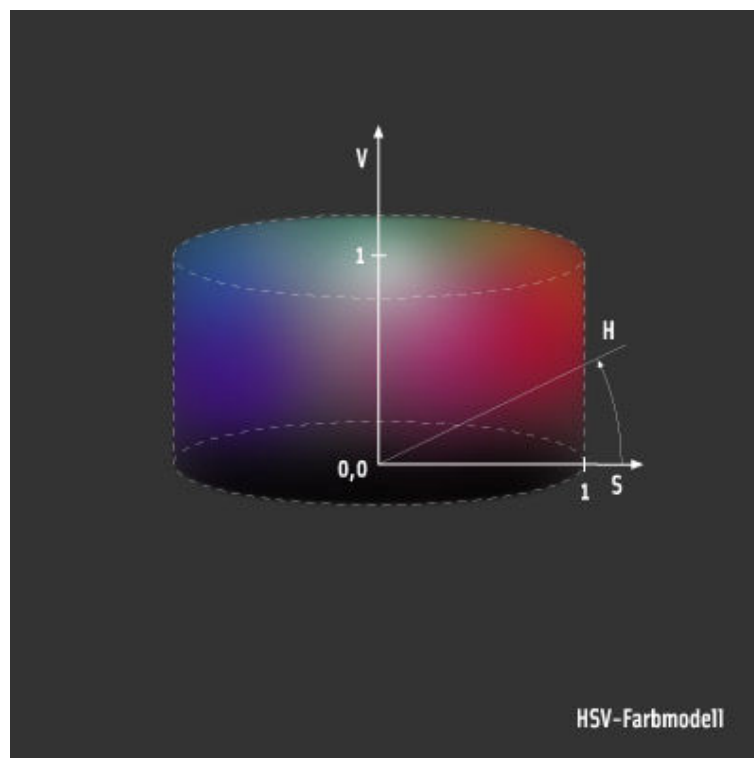


Abbildung 4.2: Das HSV-Farbmodell als Zylinder

Tabelle 4.2: Verwendete Materialien und ihre Eigenschaften

Material		Rauheit Ra ¹	Rauheit Rz ²	Farbe (H/S/V)
	0,80	0,11 μm	0,75 μm	
AlMg4,5Mn	1,00	0,46 μm	2,62 μm	
	1,20	0,07 μm	0,48 μm	grün-blau (148/0,08/0,51)
	0,80	0,10 μm	0,80 μm	
AlSi5	1,00	0,50 μm	2,11 μm	
	1,20	0,07 μm	0,50 μm	grün-blau (161/0,11/0,52)
	0,80	0,11 μm	0,92 μm	
Al99,5	1,00	0,45 μm	2,51 μm	
	1,20	0,05 μm	0,46 μm	grün-blau (154/0,08/0,47)
	0,80	0,09 μm	0,98 μm	
SG-2	1,00	0,14 μm	1,94 μm	
	1,20	0,06 μm	0,98 μm	gelb-grün (68/0,31/0,46)
	0,80	0,05 μm	0,47 μm	
S 6025	1,00	0,22 μm	1,53 μm	
	1,20	0,08 μm	1,01 μm	grün-blau (171/0,11/0,44)
	0,80	0,21 μm	1,64 μm	
SG-CuAl8	1,00	0,27 μm	1,95 μm	
	1,20	0,18 μm	1,43 μm	orange (53/0,28/0,54)
	0,80	0,14 μm	1,25 μm	
SG-CuSi3	1,00	0,19 μm	2,01 μm	
	1,20	0,10 μm	1,05 μm	orange-gelb (56/0,33/0,53)
	0,80	0,07 μm	0,85 μm	
CN 22/9	1,00	0,04 μm	0,39 μm	
	1,20	0,27 μm	1,86 μm	grün-blau (172/0,08/0,49)
SG-CuSi3MnAl	1,20	0,19 μm	1,54 μm	gelb-grün (66/0,30/0,40)
EAS 4 M-FD	1,20	0,06 μm	0,80 μm	grün (120/0,09/0,55)

¹Arithmetischer Mittenrauwert nach DIN EN ISO 4287²Maximale Rauheitsprofilhöhe nach DIN EN ISO 4287

4.3 Untersuchung verschiedener Lichtquellen

4.3.1 Glühlampenlicht/Kaltlicht

Die Beleuchtungsfrequenz beträgt 50 Hz. Damit ist die Verwendung von Glüh- und Kaltlichtlampen von vornherein ausgeschlossen. Glühlampen glimmen nach dem Ausschalten zu lange nach und führen zu einer Beleuchtung über den Lichtimpuls hinaus. Kaltlichtlampen haben eine eigene Taktfrequenz, welche mit der Impulsfrequenz interferieren würde. Auch sind Kaltlichtlampen zu träge in der Einschaltzeit.

4.3.2 Diodenlaserlicht

Zum Versuch wurde ein Lasermodul OLSH503P verwendet, welches mit rotem Licht (Wellenlänge 650 nm) arbeitet. Es sind leider nur Einschaltzeiten von 1 ms und mehr möglich, da der Laser zu träge auf kürzere Impulse reagiert. Um diese Belichtungszeiten zu erreichen, muss der Kondensator C4 auf 0.047 μF und der Widerstand R11 auf ca 3 k Ω auf der Triggerschaltung erhöht werden. Dieses RC-Glied stellt die Zeit des Lichtimpulses ein. Außerdem muss der Laser stark defokussiert werden, um die nötige Fläche von ca. 5x2 mm² auszuleuchten. Auf den Aufnahmen dieser Bilder kann man deutlich das für Laserlicht typische Speckle-Muster erkennen, welches durch die Reflexion von kohärentem Licht auf unregelmäßigen Oberflächen immer entsteht.

Die Kamera nimmt nur die hellsten Flecken dieses Musters auf, dadurch scheint die Oberfläche des Drahtes nur noch aus weit auseinander liegenden Punkten zu bestehen (s. Abbildung 4.3). Der Kontrast zum Drahtintergrund ist nicht befriedigend. Nach einer Optimierung der Beleuchtung konnten Bilder gemäß Abbildung 4.4 erzielt werden. Eine Auswertung dieser Filmsequenzen führte für alle Drahtqualitäten zu keinen reproduzierbaren Ergebnissen. Die Begründung liegt einerseits in den auftretenden Speckle-Mustern und, ebenfalls daraus resultierend, in der schlechten Detektion des Drahtumrisses.

4.3.3 Leuchtdiodenlicht

Neben den Laserdioden und den Gasbogen- und Blitzlampen sind die Leuchtdioden (kurz LED) die einzigen Lichtquellen, die den Anforderungen für den Versuchsstand gerecht werden können. Hierbei dienen 20 kreisförmig angeordnete Leuchtdioden, welche so geneigt sind, dass sie ihr Licht auf einen Punkt projizieren (s. Abbildung 4.6) als Beleuchtung der Aufnahmeszene.

Leuchtdioden der Farbe Rot erzielen die besten Helligkeiten auf den Bildern des T.I.M.M., da der CCD-Sensor bei langwelligem Licht ein besseres Ansprechverhalten aufweist (s. Abbildung 4.5). Für manche Bildverarbeitungsaufgaben kann sogar Licht im nahen Infrarotbereich für die Beleuchtung verwendet werden. Die meisten Chips haben ihre größte Aufnahmefähigkeit im Bereich zwischen 600 nm und 800 nm. (Quelle: <http://www.theta-system.de/datasheet/WEBccdPrimer2.pdf>).

Die Lichtsteuerung (s. Abschnitt 3.2.5) gibt Impulse von 100 μs und 50 μs aus. Zum Nachweis der Reaktionsschnelligkeit der Leuchtdioden wurde mit einer Hochgeschwindigkeitskamera das Verhalten einer LED des Lichttringes aufgenommen (s. Tabelle 4.3). Diese Kamera ist in der Lage 8000 Bilder in einer Sekunde zu erfassen. Ein Bild füllt also ein Zeitfenster von 125 μs . Diese Zeitauflösung ist zu grob für die zu untersuchenden Lichtimpulse, aber eine qualitative Aussage über die Fähigkeit der Leuchtdioden, die geforderten Belichtungszeiten bereitzustellen, kann dennoch getroffen werden.

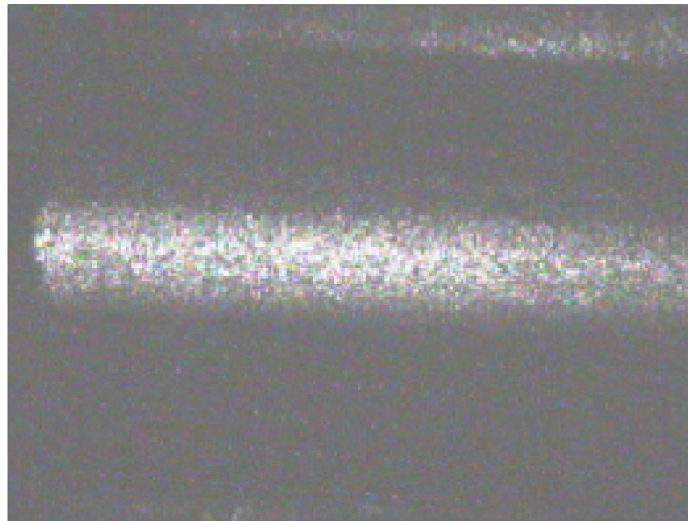


Abbildung 4.3: Beleuchtung des Kameraraumes mit einer Laserdiode.
Belichtungszeit: 1 ms

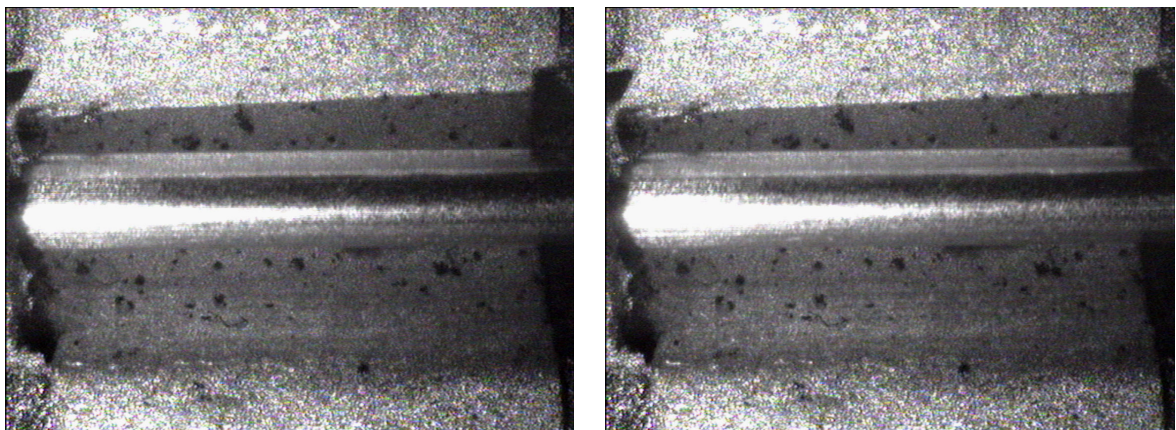


Abbildung 4.4: Aufeinander folgende Bilder bei Beleuchtung mit Laserlicht
(Draht: SG-2, $d = 1,2 \text{ mm}$)

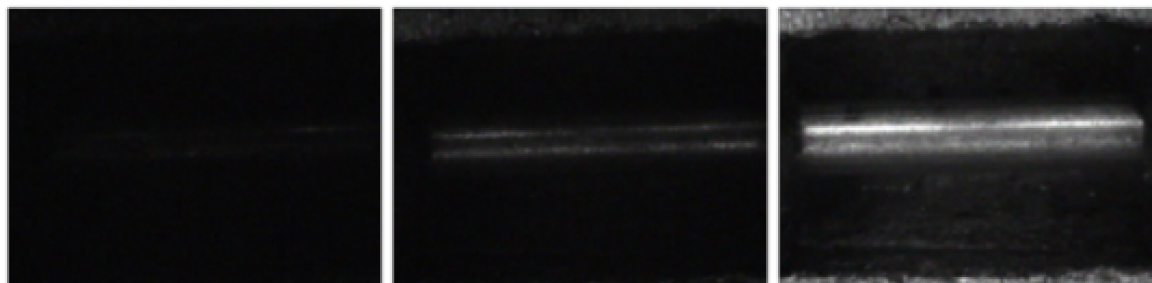


Abbildung 4.5: Belichtung der Szene mit verschiedenen farbigen Leuchtdioden (von links nach rechts: grün, gelb, rot)

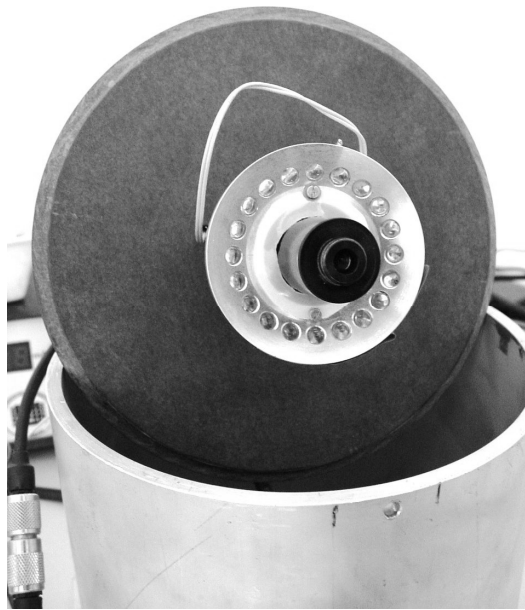


Abbildung 4.6: CCD-Kamera mit LED-Lichtring im Deckel des Versuchsstandes

Der $50\ \mu\text{s}$ -Impuls bewirkt Belichtungszeiten von $125\ \mu\text{s}$, der $100\ \mu\text{s}$ -Impuls belichtet 2 Bilder ($250\ \mu\text{s}$). Wie in Tabelle 4.3 erkennbar ist, haben die verwendeten Leuchtdioden einen relativ stark abgegrenzten Lichtaustrittsbereich. Bei stark reflektierenden Drahtoberflächen zeichnet sich dieser Lichtfleck auf der Kamera ab und verfälscht somit die Bildauswertung. Dem Draht wird dadurch ein virtueller Stillstand aufgeprägt. Um diesen Effekt zu verringern, wurden Milchglasfilter eingesetzt. Dadurch wird die Lichtintensität wieder etwas verringert, was sich aber über eine geringfügige Verlängerung der Belichtungszeit ausgleichen lässt.

Tabelle 4.3: Lichtimpulslänge, aufgenommen mit einer Hochgeschwindigkeitskamera

Zeit [μs]	Impulslänge	
	$100\ \mu\text{s}$	$50\ \mu\text{s}$
125		
250		
375		
500		
625		

5 Mathematische Algorithmen zur Versuchsauswertung

5.1 Allgemeines zu den Aufbereitungs- und Auswertalgorithmen

5.1.1 Aufbereitungsalgorithmen

Die Aufbereitungsalgorithmen sollen eine gute Auswertbarkeit der Daten sicherstellen. Dies umfasst sowohl die Datenreduktion, als auch die Korrektur von Abweichungen des Messobjektes im Bildfeld von der Optimalposition sowie Filterung des Bildes zur Beseitigung von Beleuchtungseinflüssen. Dazu sind folgende Schritte auszuführen und softwaremäßig umzusetzen:

- Beseitigung der vertikalen Bewegungseinflüsse
- Beseitigung von Winkelabweichungen (Bildrotation)
- Farbtiefenumwandlungen (Umrechnung RGB-Graustufen)
- Entwicklung digitaler Filter (soft-thresholding)

Signale lassen sich nach den verschiedensten Algorithmen so aufbereiten, dass ihre Auswertbarkeit erreicht bzw. verbessert und die Verlässlichkeit des Ergebnisses erhöht wird. Ein sehr weites Anwendungsspektrum, sowohl im 1D- als auch 2D-Bereich, bieten dabei die Wavelets [18, 4, 3, 34, 35]. Nähere Erläuterungen dazu sind im Abschnitt 5.1.3 zu finden.

5.1.2 Auswertalgorithmen

Verschiedene Auswertverfahren sind in Bezug auf die zu erreichende Genauigkeit sowie ihre Echtzeitfähigkeit zu testen. Folgende Verfahren bieten sich an:

- Messmethoden nach dem Dopplerprinzip
- Messmethoden nach dem Prinzip der Ortsfrequenzfilterung
- Messmethoden nach verschiedenen Korrelationsprinzipien

Auf die beiden erstgenannten Messprinzipien soll hier nicht näher eingegangen werden, da sie in Anbetracht des Zweckes des Versuchsstandes einen zu hohen Mess- und Geräteaufwand mit sich führen würden. Diese Prinzipien sind ausführlich in der Literatur beschrieben z. B. [27, 28]. Der beschriebene Versuchsstand soll nach dem Prinzip der Flächendifferenz und/oder Kreuzkorrelation arbeiten. Diese Verfahren bieten sich aus zwei Gründen an. Es ist erstens nur eine eindimensionale Bewegung zu untersuchen und langsame Geschwindigkeiten sowie der Stillstand des Objektes können sicher ermittelt werden. Zweitens ist die benötigte Gerätetechnik kostengünstig, einfach und robust.

Es kommen Transformations- und Kompressionsalgorithmen auf Basis der Wavelets sowie der Fouriertransformation zur Anwendung, wobei die Minimierung der Rechenzeit im Vordergrund steht.

Korrelation und Flächendifferenzbildung

Geht man davon aus, dass sich die Merkmale in einer Pixelreihe fortpflanzen (durch Aufbereitungsalgorithmen sicherzustellen), dann ist der Verschiebevektor nach der Ermittlung seines Betrages (Verschiebewert s) darstellbar. Um den Verschiebewert zu ermitteln lässt sich die Korrelation bzw. Flächendifferenzbildung anwenden.

Nachdem diese Methoden für die Ermittlung in allen Pixelreihen angewandt wurden, kann zur Erhöhung der statistischen Sicherheit eine Mittelwertbildung durchgeführt werden. Nun lässt sich durch geeignete Verfahren das globale Extrema der entstandenen Funktion ermitteln, dessen Abszissenwert dem Verschiebewert s in Pixel entspricht. Bei bekannter Pixelauflösung a und konstanter Bildwiederholfrequenz f_B ergibt sich die Drahtvorschubgeschwindigkeit v_D zu:

$$v_D = s \cdot a \cdot f_B \quad (5.1)$$

5.1.3 Effekte durch die Nutzung von Wavelets

Komplette Berechnung im Waveletbereich

Ohne Kompression erfordert die Berechnung der Flächendifferenz insgesamt $O(N^3)$ Operationen. Hat das ursprüngliche Signal N^2 Koeffizienten, so bleiben nach der Datenkompression per Wavelettransformation grob $O(N)$ Koeffizienten übrig. Mit den Tensor-Produkt-Wavelet erreicht man sogar eine Reduktion auf $O(N^{0.5})$ bzw. $O(\log N)$. Will man diese Datenreduktion ausnutzen, muss die Berechnung der Korrelation bzw. Flächendifferenz im Waveletbereich erfolgen. Allerdings verliert der im Zeitbereich genutzte Verschiebungs- oder Translationsoperator, der ein Signal um eine gewisse Anzahl von Indices bzw. Pixel weiter verschiebt, im Waveletbereich seine einfache Struktur. Ohne eine geschickte Behandlung des Translationsoperators geht demnach die durch die Datenkompression gewonnene Effizienz wieder vollständig oder zumindest teilweise verloren.

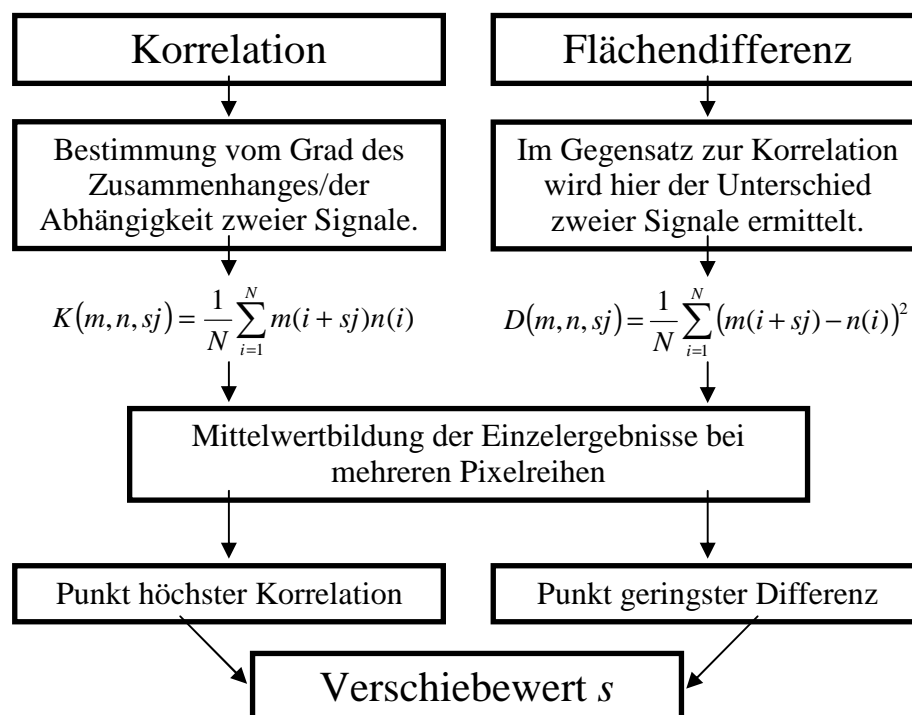


Abbildung 5.1: Prinzip der Ermittlung des Verschiebewertes s

Teilbereiche durch Wavelettransformierte effizienter lösen

Es werden hier verschiedene Varianten vorgeschlagen, dieses Problem zu behandeln. Die erste Möglichkeit besteht darin, quer zur Vorschubrichtung eine eindimensionale Wavelettransformation und Datenkompression vorzunehmen, und längs der Vorschubrichtung unter Zuhilfenahme des Faltungssatzes und der Fast Fouriertransformation (FFT) die Kreuz-Korrelation zu maximieren. Dies verzichtet allerdings auf die Möglichkeit der Datenreduktion per Wavelettransformation in Vorschubrichtung und erfordert für die Berechnung der Korrelation $O(N \log N)$ Operationen.

Die zweite Möglichkeit besteht in einem hierarchischen Vorgehen, welches die verschiedenen Skalen der Wavelettransformierten nutzt. Man beginnt das Problem auf einer groben Skala zu lösen und versucht die genaue Verschiebung durch lokale Korrekturen unter Hinzunahme jeweils der nächsten Skala zu bestimmen. Hierfür erwartet man eine Komplexität zwischen $O(N^{0.5})$ und $O(N)$. Da dieser Algorithmus aber komplizierter als der erste ist, bleibt fraglich, ob dieses theoretisch überlegene Konzept in der praktischen Implementierung tatsächlich schneller ist.

5.2 Beschreibung der mathematischen Vorgehensweise

5.2.1 Prinzip im Überblick

Es wird zunächst ein Paar von Bildern welche aus einem zusammengesetzten Vollbild gewonnen werden betrachtet (Halbbildauswertung s. Abschnitt 3.2.4). Im nächsten Schritt muss man die Teile der Bilder ausschneiden, die nur die Drahtoberfläche enthalten. Das reduziert die Menge der Daten, die bearbeitet werden müssen, und entfernt den störenden Hintergrund. Der folgende Schritt besteht darin, die Teile der Bilder zu lokalisieren, die zueinander ähnlich sind. Dafür wird eine so genannte „Ähnlichkeitsfunktion“ benutzt, die einen quantitativen Wert ermittelt, der angibt, wie verschiedene Teile des ersten Bildes zu einem fixierten Teil des zweiten Bildes ähnlich sind. Die berechnete Funktion wird dann auf die wahrscheinlichsten Werte des Vorschubs untersucht. Um jetzt die Geschwindigkeit zu berechnen, muss man die Auflösung des Systems kennen. Dafür dient die Kalibrierprozedur, die nur einmal vor der Bearbeitung abgerufen werden muss. Der gesamte Algorithmus ist in Abbildung 5.2 dargestellt. Nachfolgend einige Anmerkungen zu den zu verarbeitenden Bildern:

- Bei vielen Drähten ist es praktisch unmöglich, den Vorschub mit bloßen Augen zu beobachten, sofern es kein gut erkennbares Merkmal auf der Drahtoberfläche gibt.
- Die Antriebsrollen können Muster auf dem Draht hinterlassen, die man nicht sieht, welche aber für die hier vorgeschlagene Methode von Bedeutung sind.
- Die Beleuchtung ist inhomogen. Dieses Problem konnte nicht gelöst werden. Die inhomogene Beleuchtung führt dazu, dass die Bilder desselben Gebiets des Drahtes in verschiedenen Positionen sich enorm unterscheiden. Da die Beleuchtungsbedingungen am vorhandenen Versuchsstand nicht weiter optimierbar waren, ist es sehr schwer, eine Norm der Ähnlichkeit einzuführen. In Abbildung 5.3 sind zwei aufeinander folgende Bilder dargestellt. Im linken Bild ist die Mitte des Drahtes ähnlich beleuchtet wie das letzte Drittel des rechten Bildes.
- Auftauchen von metallischen Partikeln (s. Abbildung 5.4). Diese Partikel können sich zusätzlich während der Aufnahme bewegen. Die senkrechten Linien sind im gleichen Abstand von der linken Seite des Bildes eingezeichnet, um die Bewegung zu

verdeutlichen. Dieses Problem konnte durch die Aufbereitungsalgorithmen teilweise gelöst werden. Es ist schwieriger den Draht auf dem Bild zu finden, wenn diese Verunreinigungen auftauchen oder sich bewegen.

- Die Bewegung des Drahtes ist nicht waagrecht in der Bildebene, er kippt. Dieses Problem konnte durch die Aufbereitungsalgorithmen gelöst werden. Die Neigung des Drahtes hat das Vorzeichen gewechselt, weshalb es schwieriger ist, die ähnlichen Teile der Bilder zu lokalisieren (s. Abbildung 5.5).
- Der Draht ist extrem krumm. Dieses Problem wurde nicht gelöst. Es tritt allerdings sehr selten auf (s. Abbildung 5.6).
- Wenn die Geschwindigkeit des Drahtes so hoch ist, dass die Länge der Strecke, die der Draht zurücklegt, größer als die Länge des Gebiets ist, das aufgenommen wird, ist man nicht in der Lage den Vorschub zu berechnen. Außerdem ist nicht feststellbar, wann diese Situation eintritt.

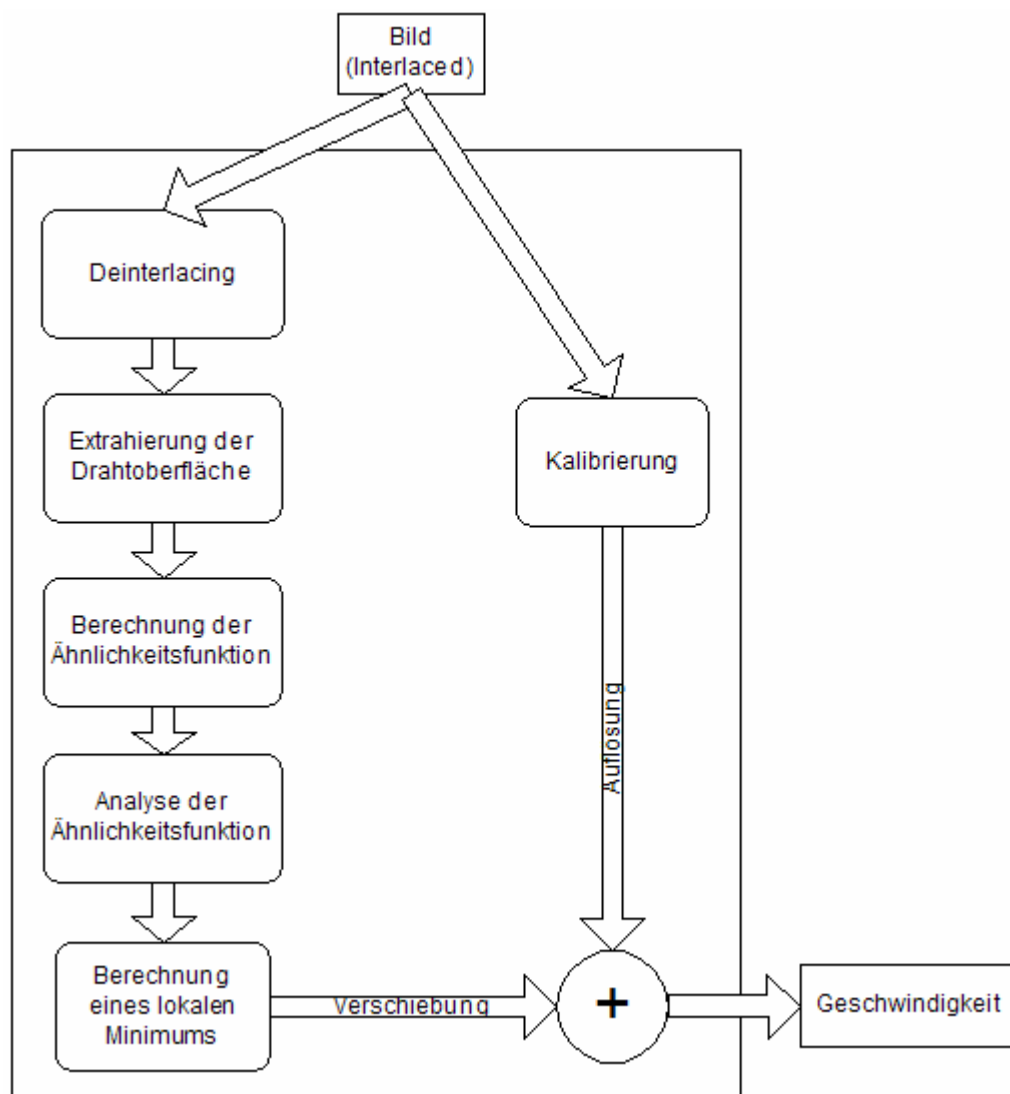


Abbildung 5.2: Schema der Verschiebungsermittlung

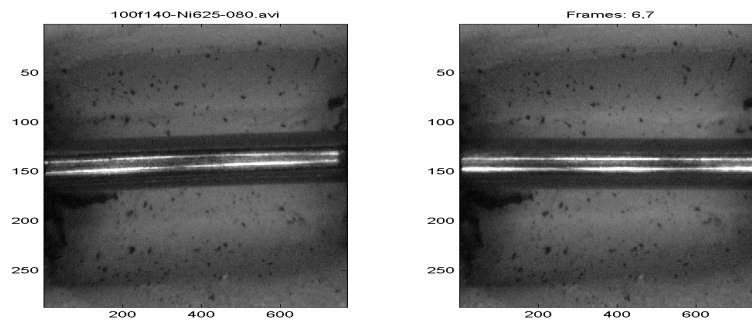


Abbildung 5.3: Aufeinander folgende Bilder mit inhomogener Beleuchtung

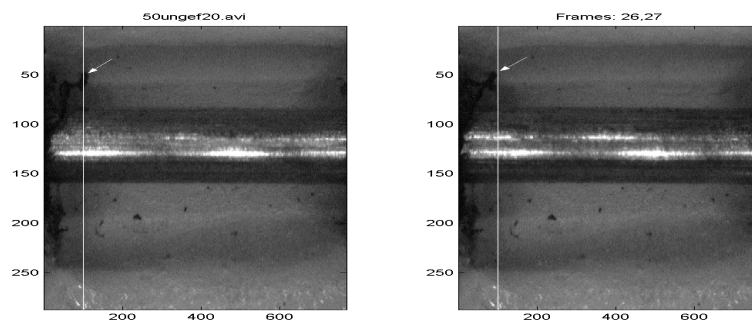


Abbildung 5.4: Sich bewegende metallische Partikel in zwei aufeinander folgenden Zeitschritten

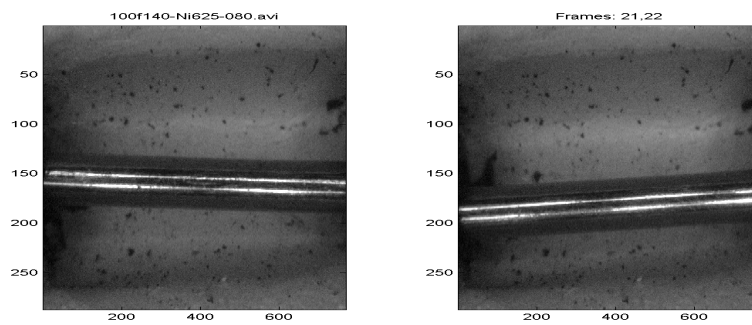


Abbildung 5.5: Darstellung der Drahtneigung in zwei aufeinander folgenden Zeitschritten

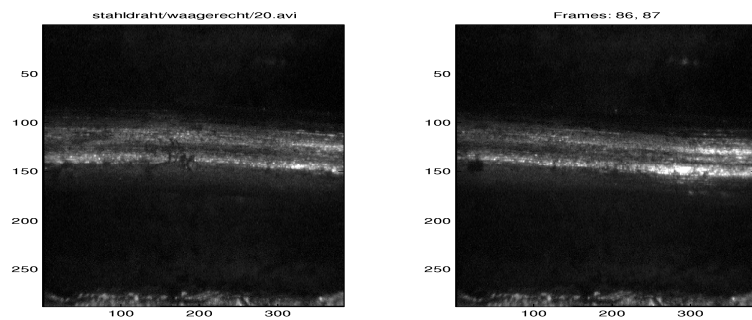


Abbildung 5.6: Bilder der Drahtkrümmung

5.2.2 Notationen im Matlab

Es wird für die Darstellung eines Bildes die Matlab-Notation benutzt. Unter einem Bild A verstehen wir eine Matrix $A(\cdot, \cdot)$ mit ganzzahligen Einträgen, wobei $A.NrOfColomns$ oder $A.Width$ die Breite von Bild A ist, und $A.NrOfLines$ oder $A.Height$ die Höhe des Bildes ist. Wir bezeichnen mit $A(i, j)$ den Eintrag der Matrix A an der Stelle (i, j) , $0 < i < A.NrOfLines + 1$, $0 < j < A.NrOfColomns + 1$. Unter $A(i_1: s_i : i_2, j_1 : s_j : j_2)$ verstehen wir eine Matrix B , die an der Stelle (k, l) den Eintrag $A(i, j)$ hat, wobei $i = i_1 + (k-1)s_i$, $j = j_1 + (l-1)s_j$:

$$B(k, l) = A(i_1 + (k-1)s_i, j_1 + (l-1)s_j), \quad k=1..[(i_2 - i_1)/s_i], \quad l=1..[(j_2 - j_1)/s_j] \quad (5.2)$$

Es sei $A(i_2: i_1, j_2: j_1) := A(i_2:1: i_1, j_2:1: j_1)$, $A(:, j_2: j_1) := A(1:A.NrOfLines, j_2:1: j_1)$. Wir können die Summe und Differenz von zwei Bildern als gewöhnliche Matrixoperationen definieren. Unter $\max(A)$ und $\min(A)$ verstehen wir den maximalen bzw. minimalen Wert der Einträge der Matrix A . Unter $\text{abs}(A)$ verstehen wir eine Matrix, deren Einträge die Beträge der Elemente der Matrix A sind. Unter der l^p - bzw. l^∞ -Norm einer Matrix A verstehen wir:

$$\|A\|_p = \left(\sum_{i=1}^{A.Width} \sum_{j=1}^{A.Height} |A(i, j)|^p \right)^{1/p}, \quad (5.3)$$

$$\|A\|_\infty = \max_{1 \leq i \leq A.Width, 1 \leq j \leq A.Height} |A(i, j)|. \quad (5.4)$$

Die lineare Faltung zweier Bilder (Matrizen) A und B wird durch folgende Faltungssumme definiert: Es sei:

$$\begin{aligned} m &= A.Width - B.Width + 1, \\ n &= A.Height - B.Height + 1, \\ bh &= B.Width, \\ bw &= B.Height. \end{aligned}$$

Dann gilt:

$$(A * B)(i, j) = \sum_{k=1}^{bh} \sum_{l=1}^{bw} A(bh + i - k, bw + j - l) B(k, l), \quad \text{wobei } 0 < i < m+1, \quad 0 < j < n+1. \quad (5.5)$$

Für ein eindimensionales Signal x (Vektor) bezeichnen wir mit $\text{decompose}(x, 'basisname', level)$ eine Wavelettransformation bis zum Niveau $level$, wobei $'basisname'$ die zugrunde liegende diskrete Waveletbasis (Haar, Beylkin, Daubechies, Coiflet, Symmlet) angibt. Für eine gegebene Matrix A bezeichnen wir mit $\text{decompose}(A, 'basisname', level)$ die entsprechende Wavelettransformation von A in x -Richtung, d. h. eine Matrix, deren Zeilen die Wavelettransformationen der Zeilen von A darstellen. Da die Wavelettransformation vom Bild A eine Matrix mit reellen Einträgen ist, werden wir auch mit der modifizierten ganzzahligen Haar-Transformation arbeiten, was die Algorithmen beschleunigt. Die modifizierte Transformation unterscheidet sich durch

die Abwesenheit eines $2^{-0.5}$ -Faktors. Für einen Vektor definieren wir einen Schritt der Transformation als

$$\begin{aligned}\tilde{x}_i &= x_{2i-1} + x_{2i}, \quad 1 \leq i \leq n, \\ \tilde{x}_i &= x_{2(i-n)-1} - x_{2(i-n)}, \quad n+1 \leq i \leq 2n,\end{aligned}\tag{5.6}$$

Im nächsten Schritt wird auf Vektor $\tilde{x} \in R^n$ genau wie oben angewendet.

Unter einem Mittelwert-Bild werden wir einen Schritt von der veränderten Haar-Transformation verstehen, wobei die Hochfrequenzkoeffizienten weggelassen sind. Die Prozeduren für die Wavelettransformation sind im Anhang 9.12 angegeben. Sie werden aus [3], [61] hergeleitet.

Wir brauchen noch die Definition eines lokalen Minimums für Funktionen, die auf einem ganzzahligen Segment definiert sind. Sei f eine Funktion, die auf einem ganzzahligen Segment $[a, b]$ definiert ist. Wir sagen, dass die Funktion f ein lokales Minimum hat, wenn eine von folgenden Bedingungen erfüllt ist:

- $x=a$ und $f(x) < f(x+1)$,
- $a < x < b$ und $f(x) < f(x+1)$ und $f(x-1) > f(x)$,
- $x=b$ und $f(x-1) > f(x)$.

5.2.3 Die Qualität des Bildes

Rauschen und effizientes Signal

Während der Experimente wurden die Teile eines Bildes einer Videosequenz, wo es keine Bewegung gab, betrachtet. Das Bild steht dabei vor einem stabilen Hintergrund (die Beleuchtungsbedingungen sind fast konstant, an diesen Stellen gibt es keine Schatten, keine sich bewegende Späne usw.). Man kann dann ein Bild B vom Hintergrund zusammenstellen, welches als Mittelwert-Bild von allen Bildern der Sequenz definiert wird (vgl. dazu Anhang 9.11). Es wird kein „Deinterlacing“ benötigt, weil es keine Bewegung gibt. Als einen quantitativen Wert des Rauschens im Bild A aus der Sequenz nimmt man die l^1 - und l^∞ -Norm der Differenz $(A-B)$, und als den Wert des effizienten Signals die Zahl:

$$\ln \frac{\max(A-B) - \min(A-B)}{\max(B) - \min(B)}.\tag{5.7}$$

Schärfe

Man versucht hier jene Information aus einem einzelnen Bild zu extrahieren, die durch Verknüpfung eines Bildpunktes mit seinen lokalen Nachbarn gewonnen werden kann. Für einen gegebenen Faltungskern L kann man als den Wert der Schärfe die Normen $\|A * L\|^1$, $\|A * L\|^\infty$ oder den Mittelwerte der Matrix $A * L$ nehmen, wobei L gegeben ist durch:

$$L_0 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \text{ oder } L_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \quad (5.8)$$

5.2.4 Algorithmus zur Kalibrierung

Für die automatische Kalibrierung kann man drei Kreise, die sich durch ihre Farbgebung gut vom Hintergrund abheben auf der inneren Oberfläche der Röhre aufbringen, wobei ihr Abstand bekannt ist. In Abbildung 5.7 ist dargestellt, wie man diese Kreise positionieren sollte. Es ist jetzt nicht schwer diese Kreise auf den Bildern zu finden und den Abstand zwischen diesen in *Pixel* zu berechnen. Die Merkmale im oberen Drittel des Bildes können für die Bestimmung der Auflösung in *x*-Richtung (Verschiebung → Geschwindigkeit) und die zwei Merkmale links können für die Kalibrierung in *y*-Richtung (Drahtstärke) verwendet werden. Die Algorithmen dazu sind bekannt und können in [63] und [65] gefunden werden. Aus diesem Grund haben wir dieses Thema nicht weiter untersucht, sondern die Auflösung des Systems als bekannt vorausgesetzt.

5.2.5 Algorithmus zum Deinterlacing

Das ursprüngliche Bild wird in zwei Bilder zerlegt, und zwar nimmt man die gerade Zeilen als das erste Bild und die ungeraden Zeilen als das zweite Bild. Bei den vorliegenden Bildern mussten die Ränder abgeschnitten werden, weil die Beleuchtung dort inhomogen ist und die erste Zeile sowie die rechte und linke Spalte von der Kamera nicht benutzt werden (sie sind immer schwarz). Von nun an werden wir nur mit einem Paar von getrennten Einzelbildern *A* und *B* arbeiten. Sei $Width=A.Width=B.Width$, $Height=A.Height=B.Height$.

5.2.6 Algorithmus zur Bestimmung der Kanten des Drahtes

Dieser Bearbeitungsschritt ist für jedes Einzelbild auszuführen. Zwei Methoden wurden untersucht.

Kantenextraktion mit Hilfe von Differenzial-Typ Operatoren

Es wurde versucht jedes Bild separat zu betrachten, und in diesem die geraden Linien zu finden. In der Literatur ist dieses Problem als „Kantenextraktion mit der Hilfe von Differenzial-Typ Operatoren“ bekannt. Um die horizontalen Linien (d. h. die Ränder des Drahtes) zu extrahieren sind folgende Hauptschritte nötig:

- Anhebung von Kanten mit der Hilfe von Faltungsoperatoren mit einem geeigneten Faltungskern, d. h. wir betrachten das gefilterte Bild $abs(A*L)$,
- für gegebene Schwelle binarisieren wir das Bild $abs(A*L)$,
- mit der Hilfe von morphologischen Operationen (z. B. „Opening“, „Closing“, „Dilatation“ und „Erosion“) versuchen wir nur die horizontalen Linien übrig bleiben zu lassen,
- jetzt versucht man die zwei Hauptlinien zu finden, die die Ränder des Drahtes sind.

Als Faltungskerne wurden folgende Operatoren benutzt:

- Laplace Operator $L_0 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$,
- Modifikation des Laplace Operators $L_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, $L_2 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$,
- Sobel Operator $S = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$.

Anstelle der ersten linearen Operation (im ersten Schritt) wurden auch nicht-lineare (Gradienten) Operatoren betrachtet, wie z. B.: $abs(A * L_i) + abs(A * S)$, $i=0..2$. Eine genauere Beschreibung kann man in [66] finden.

Nutzung von zwei aufeinander folgenden Bildern

Für ein gegebenes Paar A und B betrachtet man $abs(A-B)$. Man hofft, dass die Differenz in den Teilen des Bildes, wo es keine Bewegung gibt, fast gleich Null sein wird (s. Anhang 9.7). Da der Draht zur x -Achse parallel ist, genügt es die y -Profile zu summieren. Jetzt muss man das Gebiet finden, wo diese Summe verschieden von Null ist. Hierfür glättet man diese Funktion zunächst, und kann dann für eine gegebene Nullschwelle die Ränder des Drahtes finden.

Abbildung 5.8 zeigt links ein Interlaced-Bild, für welches man eine Summe der y -Profile von $abs(A-B)$ bildet, welche als schwarze Funktion auf der rechten Seite dargestellt ist. Die graue Funktion ist das geglättete Profil, die Gerade ist die Nullschwelle. In der Praxis zeigte diese Methode ein gutes Verhalten und ist außerdem sehr schnell. Wenn sich der Draht nicht oder nur sehr langsam bewegt, liefert diese Methode ein größeres als das benötigte Gebiet. Da die Breite des Drahtes konstant ist und die vertikale Bewegung des Drahtes beschränkt, kann man dieses Problem statistisch lösen. Man kann das laufende Ergebnis mit dem vorhergehenden vergleichen, und wenn es einen großen Unterschied gibt, nimmt man an, dass der Draht die zuvor bestimmte Breite hat und in der Mitte des oben bestimmten Gebietes liegt.

Bemerkung: Am Anfang der Forschung war der Hintergrund (d. h. die Röhre) natürlich inhomogen, weshalb die innere Oberfläche der Röhre weiß gefärbt wurde, um einen homogenen Hintergrund zu bekommen und um somit den Rand des Drahtes besser zu erkennen (man konnte ihn vorher nicht sehen). Während der Experimente wurde festgestellt, dass es besser wäre, eine schwarze Färbung zu benutzen, weil

- die auftretenden Späne unauffälliger auf dem schwarzen Hintergrund sind,
- es weniger bewegte Schatten gibt,
- die Ränder des Drahtes keine neue Information enthalten (wahrscheinlich liegen die Ränder nicht im Fokus der Kamera).

5.2.7 Algorithmus zur Berechnung der Neigung des Drahtes

Da die Ränder des Drahtes parallel sind und der Draht eine konstante Breite hat, genügt es, um die Neigung des Drahtes zu berechnen, seine Lage in horizontaler Richtung auf der linken und rechten Seite zu berechnen. Zu diesem Zweck lokalisieren wir die Ränder des Drahtes auf der linken und rechten Seite des Bildes und versuchen dann das Bild des Drahtes auf der linken Seite in einem (größeren als dem berechneten) Gebiet auf der rechten Seite bestmöglich (in l_1 -Norm) übereinander zu legen, um die Verschiebung in der horizontalen Richtung zu bestimmen. Das Gebiet darf aber nicht größer als die wirkliche Breite des Drahtes sein, der Grund dafür wird später erklärt. Wir haben ein um 20 Prozent größeres Gebiet angenommen, welches dennoch kleiner als die Breite des Drahtes war.

Sei A und B das erste und das zweite Bild aus unserem Paar. Sei $flen$ die Länge der Ebenheit des Drahtes (wir haben diesen Wert als doppelte Breite des Drahtes angenommen, dieser Wert ist kleiner als ein Drittel der Länge des gesamten Bildes). Für zwei Paare von Bildern $A(:, 1:flen)$, $B(:, 1:flen)$, und $A(:, Width-flen+1:Width)$, $B(:, Width-flen+1:Width)$ bestimmen wir die Ränder, die wir mit $UpLs$, $DownLs$, $UpRs$, $DownRs$ bezeichnen. Sei nun:

$$\begin{aligned} WidthOfWire &= (DownRs - UpRs + DownLs - UpLs) / 2 \\ UpL &= (UpLs + DownLs) / 2 - WidthOfWire / 2, \\ DownL &= (UpLs + DownLs) / 2 + WidthOfWire / 2, \\ UpR &= (UpRs + DownRs) / 2 - WidthOfWire / 2, \\ DownR &= (UpRs + DownRs) / 2 + WidthOfWire / 2. \end{aligned}$$

Jetzt legt man das Bild des Drahtes auf der linken Seite $A(UpL:DownL, 1:flen)$ in einem größeren (als dem berechneten) Gebiet auf die rechte Seite $A(UpR-WidthOfWire*0.1:DownR+WidthOfWire*0.1, Width-flen+1:Width)$ bestmöglich (in l_1 -Norm) übereinander. Hierfür berechnet man die Funktion:

$$f(s) = \| A(UpR - s : DownR - s, Width - flen + 1 : Width) - A(UpL : DownL, 1 : flen) \|_1 \quad (5.9)$$

Man hofft, dass das Minimum dieser Funktion genau für den Wert s angenommen wird, der die Neigung des Drahtes angibt. Dies ist genau dann der Fall, wenn der Definitionsbereich dieser Funktion nicht größer als die Hälfte der Breite des Drahtes ist. Zumindest ist sie ein lokales Minimum. Wie man dieses Minimum auszuwählen hat, wird in einem der nächsten Abschnitte beschrieben.

5.2.8 Algorithmus zur Berechnung der Rotation des Bildes

Wenn man die Neigung des Drahtes kennt, ist es möglich das Bild so zu drehen, dass der Draht parallel zur x -Achse ist (vgl. dazu Anhang 9.6). Folgende Methoden können betrachtet werden:

- Lineare vertikale Verschiebung: Wie man in Abbildung 5.9 erkennen kann, verschiebt man die Spalten des ursprünglichen Bildes um einen ganzzahligen Wert $k = [\alpha(j-1)]$, wobei j die Spaltennummer ist und α ein reeller Wert
- Rotation mit der Hilfe von Verschiebung: Man kann, genau wie oben, eine lineare, horizontale Verschiebung durchführen. Man kann zeigen, dass die Rotation eines Bildes äquivalent zur Operation „horizontale Verschiebung - vertikale Verschiebung -

horizontale Verschiebung“ ist. Das Schema dieser Transformation ist in Abbildung 5.10 dargestellt.

- Lineare senkrechte Verschiebung mit Mittelwertbildung: genau wie oben beschrieben, aber man versucht die Spalten kontinuierlich durch das Raster zu verschieben.
- Rotation mit der Hilfe von Verschiebung mit Mittelwertbildung: Im linken Bild der Abbildung 5.11 sieht man ein ursprüngliches Bild, für welches die Neigung gemäß Abschnitt 5.2.7 berechnet wurde. Dieses Bild wird mit Hilfe der Verschiebung gedreht. Das Ergebnis ist im rechten Bild der Abbildung 5.11 wiedergegeben.

Die ersten beiden Methoden sind nicht anwendbar, weil das Ausgangsbild nicht mehr glatt genug ist. Die anderen zwei Methoden ergaben sich aus der Arbeit [62] von A. Paeth. Das Programm für unsere Experimente wurde aus dem Artikel von Eran Yariv [<http://www.codeproject.com/bitmap/rotatebyshear.asp>] übernommen und angepasst. Der Programmtext ist in Anhang 9.10 gegeben.

Drei Kreise in kontrastreicher Farbe zum Hintergrund

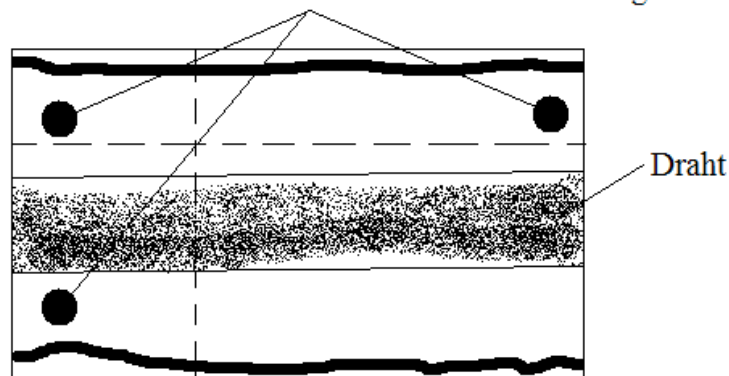


Abbildung 5.7: Lage der Kalibrierkreise

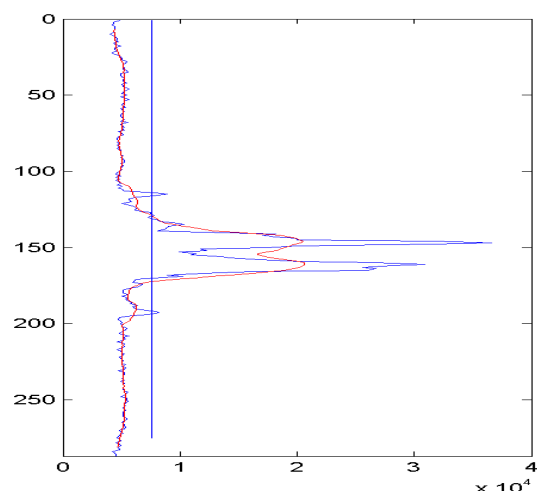
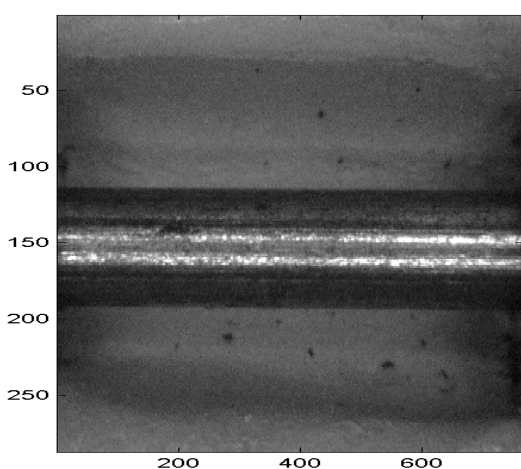


Abbildung 5.8: Vorgehensweise bei der Kantenbestimmung des Drahtes

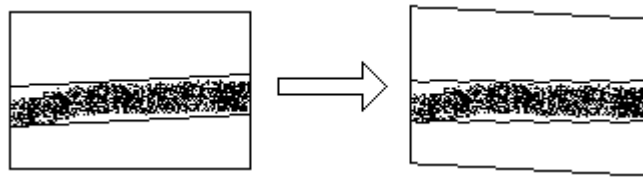


Abbildung 5.9: Darstellung der linearen vertikalen Verschiebung

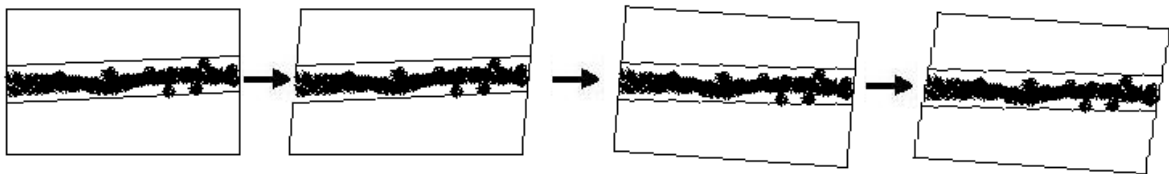


Abbildung 5.10: Die Hauptschritte bei der Drehung eines Bildes mittels ebener Verschiebungen

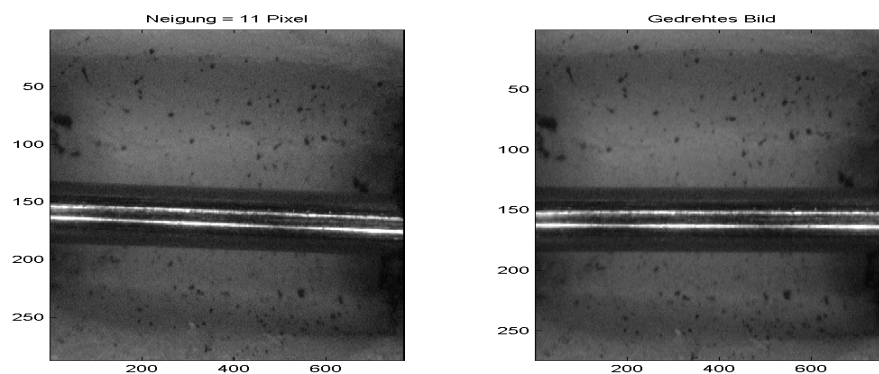


Abbildung 5.11: Drehung eines Drahtbildes

5.2.9 Allgemeines zur Berechnung der Ähnlichkeitsfunktion

Wir haben schon eine l^p -Norm für Bilder eingeführt. Als den Wert der Ähnlichkeit zweier Bilder können wir jetzt die Norm der Differenz von zwei Bildern nehmen. Die Ähnlichkeitsfunktion wird die Ähnlichkeit zwischen dem ersten Bild und der verschobenen Version des zweiten Bildes zeigen. Es gibt verschiedene Möglichkeiten diese Funktion zu definieren, und wir werden einige davon betrachten.

Wir nehmen hier an, dass der Draht sich von links nach rechts bewegt. Um die Verschiebung in die andere Richtung zu betrachten, muss man einfach das erste Bild mit dem zweiten vertauschen.

Wir nehmen weiterhin an, dass die zwei Bilder A und B die gleichen Abmessungen besitzen. Sei A das erste Bild und B das zweite. Wir können aus diesen Bildern mittels der Methode aus Abschnitt 5.2.6 jeweils die Bilder extrahieren, welche nur die Drahtoberfläche darstellen. Es sei $AW(BW)$ das Bild der Drahtoberfläche aus dem Bild $A(B)$.

5.2.10 Zyklische Verschiebung und Korrelation

Wir nehmen hier an, dass die Bewegung des Drahtes parallel zur x -Achse ist. In diesem Fall wird die Ähnlichkeitsfunktion f auf der Menge aller möglichen zyklischen Verschiebungen in x -Richtung des Bildes B , auf dem ganzzahligen Segment $[0, A.Width-1]$ definiert. Der Wert dieser Funktion für eine Verschiebung s ist die l^p -Norm der Differenz von Bild A und dem zyklisch in x -Richtung um $(-s)$ Pixel verschobenen Bild B :

$$f(s) = \| AW(:, 1:Width-s) - BW(:, s+1:Width) \|_1 + \| AW(:, Width-s+1:Width) - BW(:, 1:s) \|_1 \quad (5.10)$$

Man kann hier anstelle von f eine Korrelationsfunktion einsetzen, es ist aber offensichtlich, dass dieser Ansatz äquivalent zum Fall $p=2$ ist.

Im Fall, dass der Draht nicht parallel zur x -Achse ist, er aber auf den beiden Bildern parallel ist, kann man diese Methode durch die Einführung einer adaptiven Ähnlichkeitsfunktion modifizieren. Und zwar werden wir ein zyklisch verschobenes Bild B als ein Paar von Bildern betrachten, wobei jedes Bild separat auf die entsprechenden Teile des Bildes A übereinander gelegt wird:

$$f(s) = \min_t \| A(Up-t:Down-t, 1:Width-s) - BW(:, s+1:Width) \|_1 + \min_t \| A(Up-t:Down-t, Width-s+1:Width) - BW(:, 1:s) \|_1 \quad (5.11)$$

5.2.11 Azyklische Verschiebung

Da wir angenommen haben, dass der Draht sich von links nach rechts bewegt, bedeutet das, dass es nur jeweils einen Teil von jedem Bild gibt, welcher übereinstimmen kann. Diese Eigenschaft wurde oben nicht benutzt. Deshalb betrachten wir eine andere Ähnlichkeitsfunktion, die auch auf demselben Segment definiert ist. Die Werte dieser Funktion sind die Normen der Differenz zwischen dem linken und dem rechten Teil jedes Bildes, die für eine gegebene Verschiebung s die Länge von $A.Width-s$ haben:

$$f(s) = \min_t \| A(Up-t:Down-t, 1-Width-s) - BW(:, s+1:Width) \|_1. \quad (5.12)$$

Es ist klar, dass diese Funktion nicht monoton wachsend sein sollte, was jedoch nicht immer der Fall ist. Die Ursache dafür liegt in der inhomogenen Beleuchtung. Die Information, die diese Funktion für große Verschiebungen liefert, ist außerdem nicht so vertrauenswürdig, weil dann die überschneidende Umgebung klein ist.

5.2.12 Azyklische Verschiebung mit Fenster

Um den Einfluss des nicht monoton wachsenden Verhaltens zu beseitigen, ist es günstig eine minimale Länge der Bilder festzulegen, für die eine Aussage über ihre Ähnlichkeit gegeben werden kann. Dieser Wert der Länge ist eng verbunden mit dem oben erwähnten Wert f_{len} (Abschnitt 5.2.6), weil der Teil des Bildes der Drahtoberfläche dieser Länge hoffentlich „gerade“ und „parallel zur x -Achse“ sein wird. Die Idee ist ein Fenster (d. h.

einen Teil des Bildes der Drahtoberfläche, wo z. B. ein Merkmal liegen kann) aus dem erstem Bild zu nehmen, und eine beste Position für dieses in dem zweiten Bild zu finden.

Wir bezeichnen mit L die Länge des Fensters. Unsere Ähnlichkeitsfunktion wird für das Bild A und für das Fenster (ein Teil des Bildes B) der Länge L , auf dem ganzzahligen Segment $[0, A.Width-L]$ definiert. Der Wert der Funktion ist die l^p -Norm der Differenz zwischen dem Fenster und einem Teil des Bildes A , welches relevant zur Verschiebung s ist:

$$f(s) = \min_t \| A(Up-t : Down-t, Width-L+1-s : Width-s) - BW(:, Width-L+1 : Width) \|_1, \quad (5.13)$$

wobei wir $BW(:, Width-L+1 : Width)$ als Fenster genommen haben.

5.2.13 Azyklische Verschiebung mit Fenster und Beleuchtungskorrektur

Es wurde bereits erwähnt, dass die Beleuchtungsbedingungen sich im ganzen Bild stark ändern. Auf der anderen Seite, da die Länge des Fensters bedeutend kleiner als die Länge des gesamten Bildes ist, nehmen wir an, dass die Beleuchtungsbedingungen im Fenster homogen sind. Die Idee besteht nun darin, die obige Ähnlichkeitsfunktion so zu ändern, dass die Beleuchtungsbedingungen für jede mögliche Verschiebung s korrigiert werden:

$$f(s) = \min_{t, S, k} \| S + k * A(Up-t, Down-t, Width-L+1-s : Width-s) - BW(:, Width-L+1 : Width) \|_1. \quad (5.14)$$

5.2.14 Bemerkungen zu den Algorithmen

- Man kann die eingeführten Funktionen hinreichend effizient berechnen, weil die Funktionen unimodular sind.
- Die zwei letzten Funktionen können umgekehrt definiert werden, und zwar nehmen wir das Fenster aus dem ersten Bild und verschieben es durch das zweite Bild (die ersten zwei Funktionen werden sich fast nicht ändern).
- Das Fenster kann man in einem beliebigen Bereich wählen, die Position des Fensters bestimmt dann die minimale und maximale Geschwindigkeit g_{min} , g_{max} , die gemessen werden kann (der zweite Wert, der $(g_{max} - g_{min})$ begrenzt, ist die Länge des Fensters L).
- Das Fenster kann man so auswählen, dass es ein Merkmal enthält.
- Man legt das Fenster auf den Rand, um die maximal messbare Geschwindigkeit zu maximieren.

Bei den Untersuchungen zeigte die Ähnlichkeitsfunktion aus Abschnitt 5.2.12 das beste Verhalten hinsichtlich „Genauigkeit/Komplexität“. Deshalb ist diese Funktion zu bevorzugen (s. Anhang 9.8).

5.2.15 Beschreibung des lokalen Minimums einer diskreten Funktion

Wenn man die Ähnlichkeitsfunktion berechnet hat, muss man den Wert der wahrscheinlichen Verschiebung bestimmen. Empirisch gesehen sollte dieser Wert in der Menge aller lokalen Minima liegen. Das Hauptziel dieses Abschnittes ist, die lokalen Minima einer diskreten Funktion zu beschreiben und zu untersuchen, wie man das Geeignetste davon auszuwählen hat (vgl. dazu Anhang 9.9).

Unter der Ableitung f' einer Funktion f , die auf einem ganzzahligen Segment $[a, b]$ definiert ist, verstehen wir die folgende Funktion, die auf dem Segment $[a, b-1]+1/2$ definiert ist:

$$f'(x+1/2)=f(x+1)-f(x) \quad (5.15)$$

In diesem Kontext wird die zweite Ableitung der Funktion f auf dem Segment $[a+b, b-1]$ definiert als:

$$f''(x)=f(x-1)-2f(x)+f(x+1) \quad (5.16)$$

Wenn wir annehmen können, dass die Funktion f „glatt“ genug ist, können wir eine „genauere“ zweite Ableitung einführen, die auf dem Segment $[a+2, b-2]$ definiert ist:

$$f''(x)=-f(x-2)+16f(x-1)-30f(x)+16f(x+1)-f(x+2) \quad (5.17)$$

Es ist klar, dass wenn die erste Ableitung ihre Vorzeichen von Minus nach Plus wechselt, unsere Funktion an dieser Stelle ein lokales Minimum besitzt. Um den geeigneten Wert aus der Menge aller lokalen Minima auszuwählen, bieten sich drei Möglichkeiten an:

- Man nimmt die zweite Ableitung als den charakteristischen Wert, d. h. es wird diese Stelle ausgewählt bei der die 2. Ableitung maximal ist. Hier kann man eine der beiden möglichen Definitionen von f'' zu Grunde legen.
- Eine andere Möglichkeit besteht darin, dass wir die Funktion f wie folgt beschreiben: Es existiert eine Menge von Konstanten (gemäß Gleichung (5.18)), so dass die Funktion f auf dem Segment $[a_{i-1}, a_i]$ monoton ist, und wenn f auf $[a_{i-1}, a_i]$ monoton steigend (fallend) ist, dann muss die Funktion f auf $[a_i, a_{i+1}]$ fallend (steigend) sein. Es ist jetzt klar, dass die Menge aller lokalen Minima eine Teilmenge von $\{a_i\}_{i=1}^n$ ist. Als den charakteristischen Wert kann man die minimale Variation der Funktion f auf dem Segment $[a_{i-1}, a_{i+1}]$, wo die Funktion unimodular ist, betrachten, und zwar gemäß Gleichung (5.19)

$$a=a_0 < a_1 < \dots < a_{n-1} < a_n=b \quad (5.18)$$

$$\min\{f(a_{i+1})-f(a_i), f(a_{i-1})-f(a_i)\} \quad (5.19)$$

5.2.16 Schema des gesamten Algorithmus

Das gesamte Schema der Berechnung ist in Abbildung 5.12 dargestellt. Die Prozeduren „Berechnung der Neigung“ und „Rotation“ sind nur wichtig für die Ähnlichkeitsfunktionen aus den Abschnitten 5.2.10 und 5.2.11. Für die Ähnlichkeitsfunktionen aus den Abschnitten 5.2.12 und 5.2.13 sind sie optional, weil diese Ähnlichkeitsfunktionen die

Eigenschaft haben mit der Neigung des Drahtes umzugehen. Außerdem sollte die Rotation nur dann ausgeführt werden, wenn die Neigung größer als 3 *Pixel* ist. Die Prozedur „Rotation“ kann die Bilder entweder drehen oder vertikal verschieben.

Die Prozedur „Find Shift“ enthält zwei Hauptschritte: Berechnung von Ähnlichkeitsfunktion und Bestimmung eines lokalen Minimums dieser Funktion. Man kann eine der oben genannten Ähnlichkeitsfunktionen verwenden. Dasselbe gilt für die Bestimmung des lokalen Minimum: Man kann eine der drei vorgeschlagenen Möglichkeiten benutzen.

Die Prozedur „Statistical Control“ ist auch optional, und es gibt verschiedene Möglichkeiten diese Funktion einzuführen (s. Anhang 9.5 und 9.6).

5.2.17 Algorithmen zur Statistischen Kontrolle

Zum Zwecke der statistischen Kontrolle werden einige schon berechneten Werte gespeichert, um dann eine Aussage über die Zuverlässigkeit des neu berechneten Wertes treffen zu können. Für statistische Kontrolle der Berechnung von Kanten eignet sich die folgende Prozedur:

- Wenn die eben berechnete Breite sich vom Mittelwert der vorher berechneten Werte nicht stark unterscheidet, dann nehmen wir an, dass dieser Wert richtig ist,
- Wenn die eben berechnete Breite sich vom Mittelwert der vorher berechneten Werte stark unterscheidet, dann nehmen wir als Breite den Mittelwert der vorher berechneten Werte, und nehmen an, dass der Draht in der Mitte des eben berechneten Gebietes liegt.

Da die Berechnung der Neigung auf der Bestimmung von Kanten basiert, können wir sie ebenfalls wie oben beschrieben kontrollieren. Für die Verschiebungsberechnung kann man die statistische Kontrolle nicht so leicht einsetzen, weil eine schnelle Veränderung der Geschwindigkeit auftreten kann.

5.2.18 Beschleunigung des Algorithmus mit Hilfe der Wavelet Transformation

Es wird die folgende Idee benutzt, um den Algorithmus zu beschleunigen: Wir bilden aus den ursprünglichen Bildern „Mittelwert-Bilder“ (in x -Richtung) und versuchen für diese verkleinerten Bilder den Algorithmus anzuwenden. Man erhält eine Näherungslösung, die man mit Hilfe der ursprünglichen Bilder verbessern kann. Natürlich kann diese Methode mehrmals angewendet werden. Um ein „Mittelwert-Bild“ zu erhalten, kann man eine diskrete Wavelet Transformation verwenden. Das Wort „Mittelwert-Bild“ bezeichnet hier das Bild, das aus den niedrigen Frequenzkoeffizienten zusammengestellt ist.

Während der Untersuchungen ist die folgende Frage aufgetreten, welche unbeantwortet blieb: wie klein müssen die Koeffizienten von hohen Frequenzen sein, so dass die Ähnlichkeitsfunktion für die „Mittelwert-Bilder“ an etwa derselben Stelle das lokale Minimum besitzt, wie die Ähnlichkeitsfunktion der ursprünglichen Bilder. Für diesen Wert wurden die l^1 - und l^∞ -Norm angenommen.

Es wurden verschiedene Waveletbasen (Haar, Beylkin, Coiflet, Daubechies und Symmlet) untersucht. Dabei zeigte die Haar-Basis hinsichtlich obiger Eigenschaft das beste Verhalten. Da wir keine Koeffizienten der hohen Frequenzen brauchen, lässt sich die Berechnung schneller ausführen.

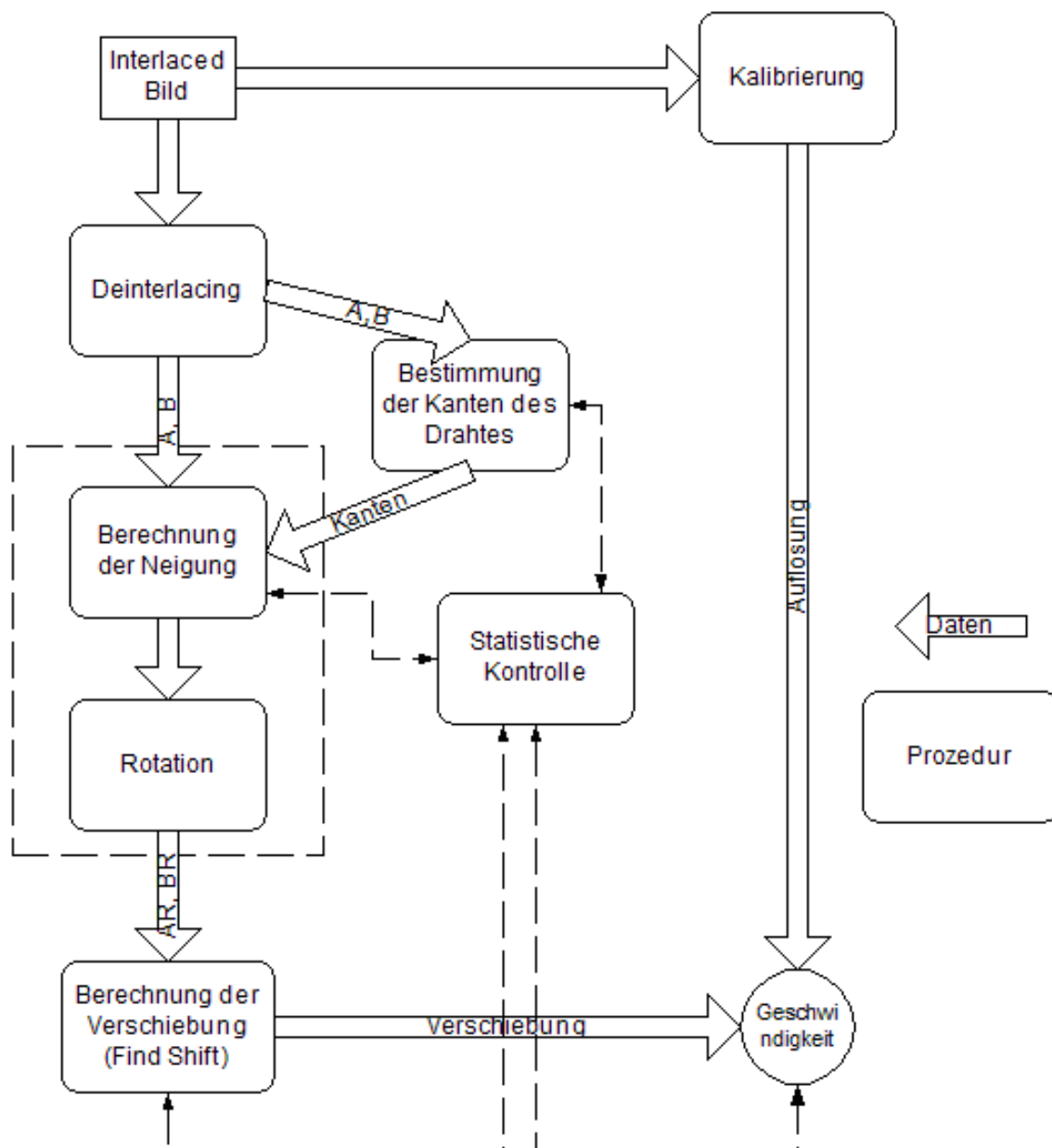


Abbildung 5.12: Schema des unbeschleunigten Algorithmus

5.2.19 Beschleunigung mit Hilfe der Mittelwert-Transformation in y-Richtung

Die andere Möglichkeit, die Anzahl der Informationen zu reduzieren, besteht darin, vor der Bearbeitung, die Bilder in y-Richtung zu komprimieren (z. B. durch Mittelwertbildung). Das heißt, wenn wir den Draht im Bild gefunden haben, können wir jetzt mit in y-Richtung verkleinerten Bildern arbeiten.

5.2.20 Schema des Schnellen Algorithmus

Auf Grund unserer Ergebnisse haben wir uns entschieden dreimal die in Abschnitt 5.2.18 beschriebene Prozedur anzuwenden. Man kann somit den schnellen Algorithmus wie in Abbildung 5.13 gezeigt durchführen.

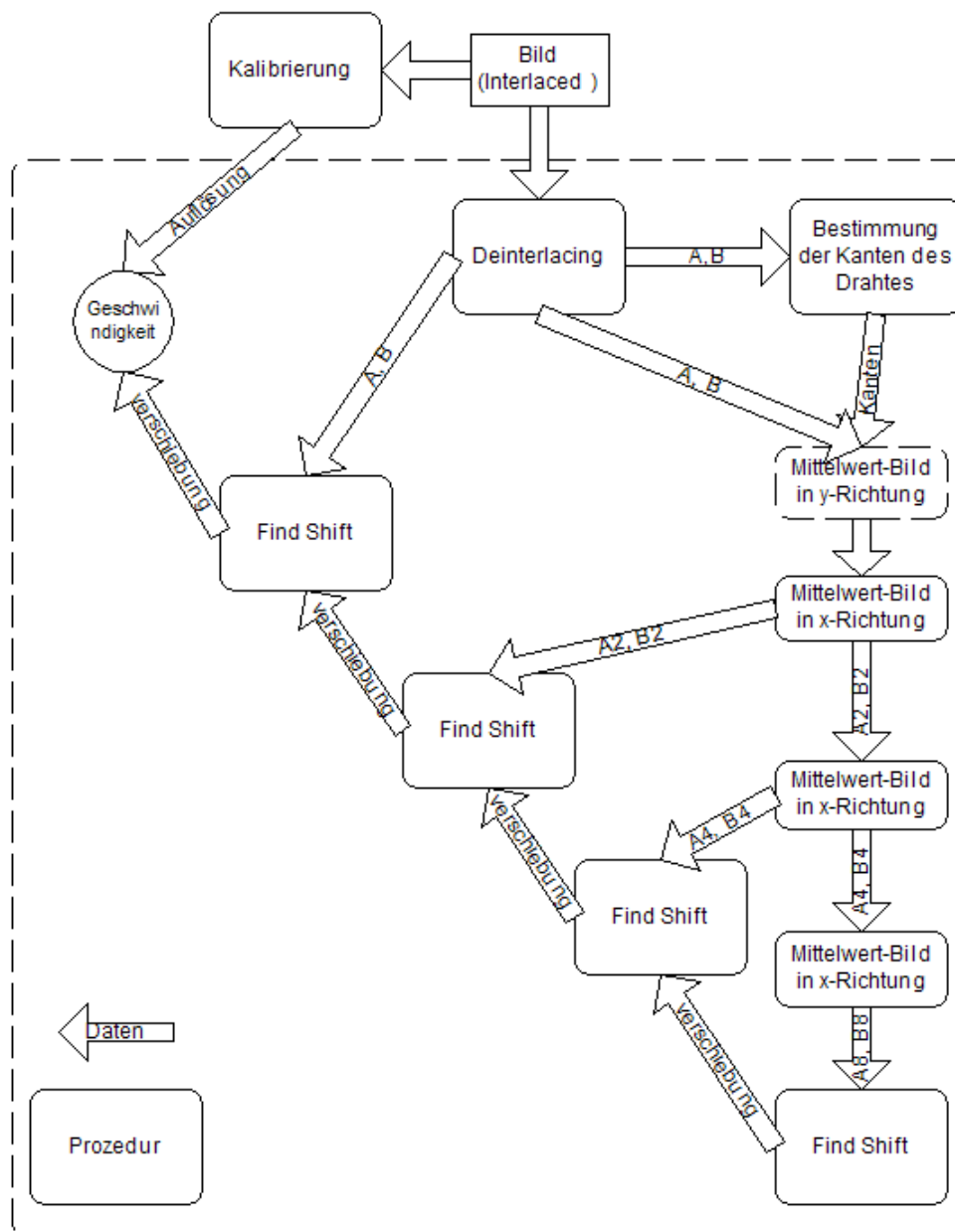


Abbildung 5.13: Blockdiagramm des schnellen Algorithmus

5.2.21 Realisierung

Alle Prozeduren, die oben beschrieben worden sind, sind als Matlab-Programme umgesetzt worden. Ein Teil von ihnen wurde nur für die Untersuchungen benötigt, deshalb haben wir für diesen Teil keinen c++ Programmtext aufgestellt. Den c++ Programmtext kann man entweder unter Windows oder Linux kompilieren (als eine DLL (Dynamically Linked Library) mit Matlab Abrufvereinbarung). Man kann natürlich denselben Programmtext ohne Matlab einsetzen.

Die Videosequenzen wurden in Matlab in der Form von AVI Files mit 5-5-5 RGB Bildern bearbeitet.

6 Versuchsdurchführung Teil 2

6.1 Aufnahme und Auswertung verschiedener Drahtqualitäten

6.1.1 Allgemeine Hinweise zur praktischen Durchführung der Versuche

Die vorhandene Anlagenkonfiguration lässt eine Aufzeichnungsgeschwindigkeit von 8,333 *fps* zu. Dies entspricht einem Bild alle 120 *ms*. Für die Auswertung wurde jeweils eine Filmsequenz von 41 Bildern (ca. 5 *s*) aufgezeichnet. Nach Gleichung (3.4) beträgt die maximal zu detektierende Drahtgeschwindigkeit theoretisch 15 *m/min* (768 *Pixel* im Bildausschnitt von 5 *mm*). Praktisch muss es eine Überlappung der Bildausschnitte geben. Die Versuche für die vorhandene Anlagenkonfiguration haben gezeigt, dass diese ca. 1/3 des Bildes betragen sollte, um eine korrekte Berechnung der Geschwindigkeit aus den Bildern abzusichern. Es stehen daher ca. 500 *Pixel* zur Verfügung, was einer maximalen Drahtvorschubgeschwindigkeit von ca. 10 *m/min* entspricht.

Die verwendete Kamera wies zudem in allen Randbereichen unsauber belichtete *Pixel* auf, welche vor der Auswertung entfernt werden mussten. Es kam daher zu einem geringen Informationsverlust, welche die minimal detektierbare Drahtvorschubgeschwindigkeit von ca. 1 *m/min* begründet (vgl. dazu Abschnitt 5.2.5).

In Tabelle 6.1 sind die Sollwerte (20, 40, 60) der durchgeführten Versuche und darüber hinaus die eben beschriebenen Grenzwerte aufgelistet.

Tabelle 6.1: Sollwerte und wichtige Grenzwerte der durchgeführten Versuche und ihre Umrechnung in andere Einheiten

Sollwert [bs2sx] ¹	Sollwert [m/min]	Sollwert [Pixel/20 ms]
1	0,14	7
10	1,42	73
20	2,84	146
40	5,69	292
60	8,53	438
70	9,95	511
100	14,22	730

¹ Die eigentliche Maßeinheit ist „Inkrement pro Programmdurchlauf“. Näheres dazu s. ANHANG Abschnitt 9.2

6.1.2 Hinweise zu den Werkstoffen

Aluminiumwerkstoffe

Alle verwendeten Aluminiumwerkstoffe besitzen eine grün-blaue Farbe und ihre Oberfläche ist hoch glänzend. Die Drähte der Durchmesser 1,0 mm besitzen eine höhere Oberflächenrauheit. Auf Grund der relativ schlechten Ergebnisse mit den Drähten dieses Durchmessers ist die Annahme widerlegt, dass verstärkte Strukturen auf der Drahtoberfläche zu besseren Ergebnissen führen könnten. Nur in ca. 86 % der 369 Messungen konnten korrekte Ergebnisse erzielt werden. Das stimmt mit den vorliegenden Erfahrungen überein, welche den Aluminiumwerkstoffen eine sehr schlechte Eignung für den Einsatz optischer Messverfahren bescheinigen. Starke Reflexionen der hellblanken Oberfläche könnten Ursache für diese Probleme sein. Andererseits ist es denkbar, dass auf Grund der leichten Deformationsfähigkeit dieser Drähte der als Sollwert angenommene Zahlenwert „falsch“ ist (s. dazu Abschnitt 7).

Stahlwerkstoffe

Die Oberfläche der verkupferten Stahlzusatzwerkstoffe entspricht einer gelb-grünen Farbe. Dadurch sind bessere Ergebnisse zu erwarten. Bei 99 % der 369 Messungen wurde die korrekte Geschwindigkeit detektiert. Auch hier besitzt der Draht des Durchmessers 1,0 mm die höchste Rauigkeit. Leichte Probleme bereiteten allerdings die Drähte des 0,8er Durchmessers.

Kupferwerkstoffe

Alle Kupferdrähte besitzen naturgemäß eine vom Orangen geprägte Farbe. Sowohl bei den Massiv- als auch den Fülldrähten lassen sich die Geschwindigkeiten mit 100 %-iger Sicherheit bestimmen. Es wurden insgesamt 369 Messungen durchgeführt.

Nickel- und Sonderwerkstoffe

Diese Werkstoffe besitzen überwiegend grüne bis blaue Oberflächenfarben mittlerer bis starker Intensität. Ihre Eignung für die optische Messtechnik entspricht in etwa der von Aluminiumwerkstoffen. Die Trefferquote liegt für Nickel bei 90 % (820 Messwerte) und für die Sonderwerkstoffe bei 83 % (369 Messwerte). Problematisch sind hier die Drähte der 0,8er und 1,0er Durchmesser. Hier können ebenfalls die starken Reflexionen der hellblanken Oberfläche Ursache für diese Probleme sein. Andererseits ist es denkbar, dass auf Grund der leichten Deformationsfähigkeit dieser Drähte der als Sollwert angenommene Zahlenwert „falsch“ ist (s. dazu Abschnitt 7).

6.1.3 Zusammenfassende Darstellung der Versuche

Tabelle 6.2 spiegelt die Ergebnisse der Versuchsreihen mit dem Sollwert 2,84 m/min wieder. Insgesamt wurden 861 Messungen durchgeführt, wobei 52 Bildkombinationen aussortiert wurden. Der maximale relative Fehler lag dabei deutlich unter 10 %, wobei Kupferdrähte exzellente Ergebnisse liefern. Durch Abbildung 6.1 wird der beschriebene Sachverhalt deutlich wiedergegeben. Der Sollwert ist unter Beachtung der auftretenden Fehlertoleranz immer bestimmbar.

Tabelle 6.2: Ergebnisse für den Sollwert 2,84 m/min

Werkstoff	Alu	Stahl	Kupfer	Ni- und Sonderw.
Istwerte im Mittel [m/min]	2,67	2,73	2,86	2,74
Max. absoluter Fehler [m/min]	0,18	0,19	0,18	0,18
Max. relativer Fehler [%]	6,2	6,8	6,2	6,2
Anzahl Messungen gesamt	123	123	123	492
Anzahl gültige Messungen	122	123	123	441

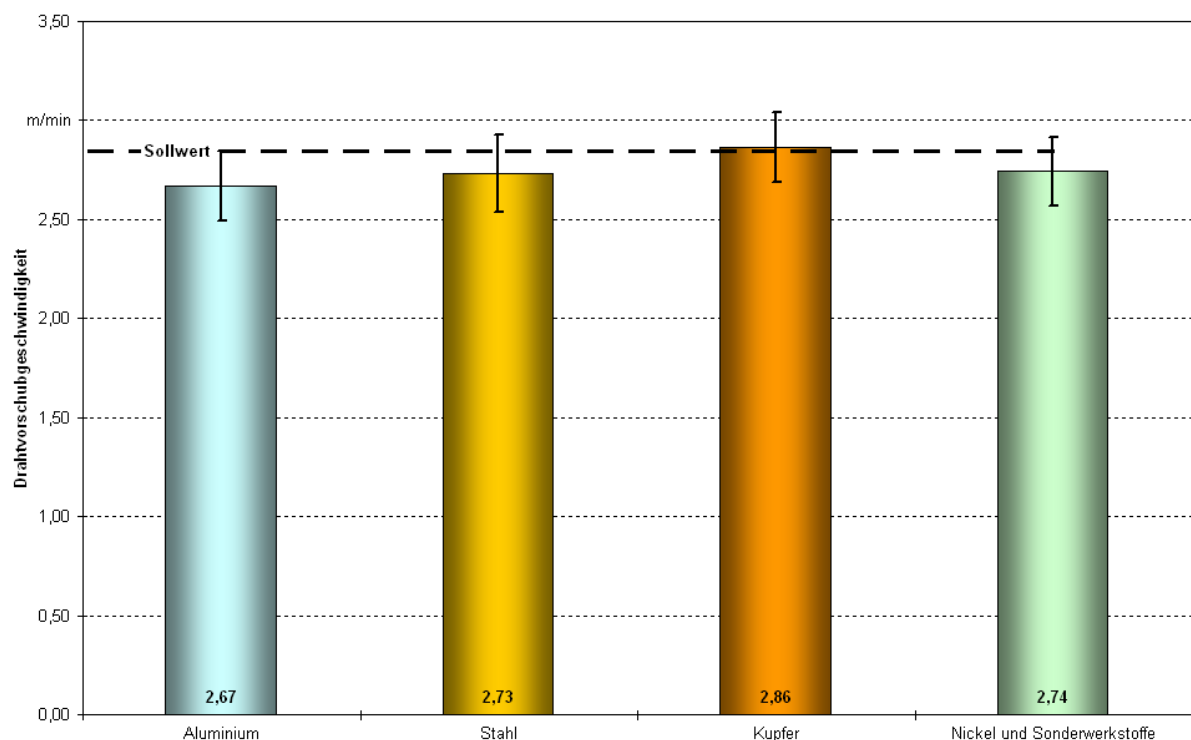


Abbildung 6.1: Ergebnisse der Versuche bei einem Sollwert von 2,84 m/min

In Tabelle 6.3 sind die Ergebnisse der Versuchsreihen mit dem Sollwert $5,69 \text{ m/min}$ dargestellt. Insgesamt wurden 861 Messungen durchgeführt, wobei 124 Bildkombinationen, wiederum vorwiegend bei Aluminium und Nickel/Sonderwerkstoffen, aussortiert wurden. Der maximale relative Fehler lag hier bei Stahl und Kupfer unter 5 %, wobei Kupferdrähte wiederum die besten Ergebnisse lieferten. Durch Abbildung 6.2 wird der beschriebene Sachverhalt nochmals deutlicher wiedergegeben. Die Stahlmessungen liefern zu geringe Drahtvorschubgeschwindigkeitswerte ($0,26 \text{ m/min}$). Auf Grund der Vielzahl der Messungen mit dementsprechend guten statistischen Ergebnissen lässt sich hier ein Rückschluss auf einen Schlupf des Drahtes ziehen. D. h. der Draht könnte tatsächlich eine etwas geringere Geschwindigkeit besessen haben.

Tabelle 6.3: Ergebnisse für den Sollwert $5,69 \text{ m/min}$

Werkstoff	Alu	Stahl	Kupfer	Ni- und Sonderw.
Istwerte im Mittel [m/min]	5,58	5,40	5,71	5,56
Max. absoluter Fehler [m/min]	0,62	0,23	0,16	0,45
Max. relativer Fehler [%]	11,0	4,1	2,7	7,9
Anzahl Messungen gesamt	123	123	123	492
Anzahl gültige Messungen	93	122	123	399

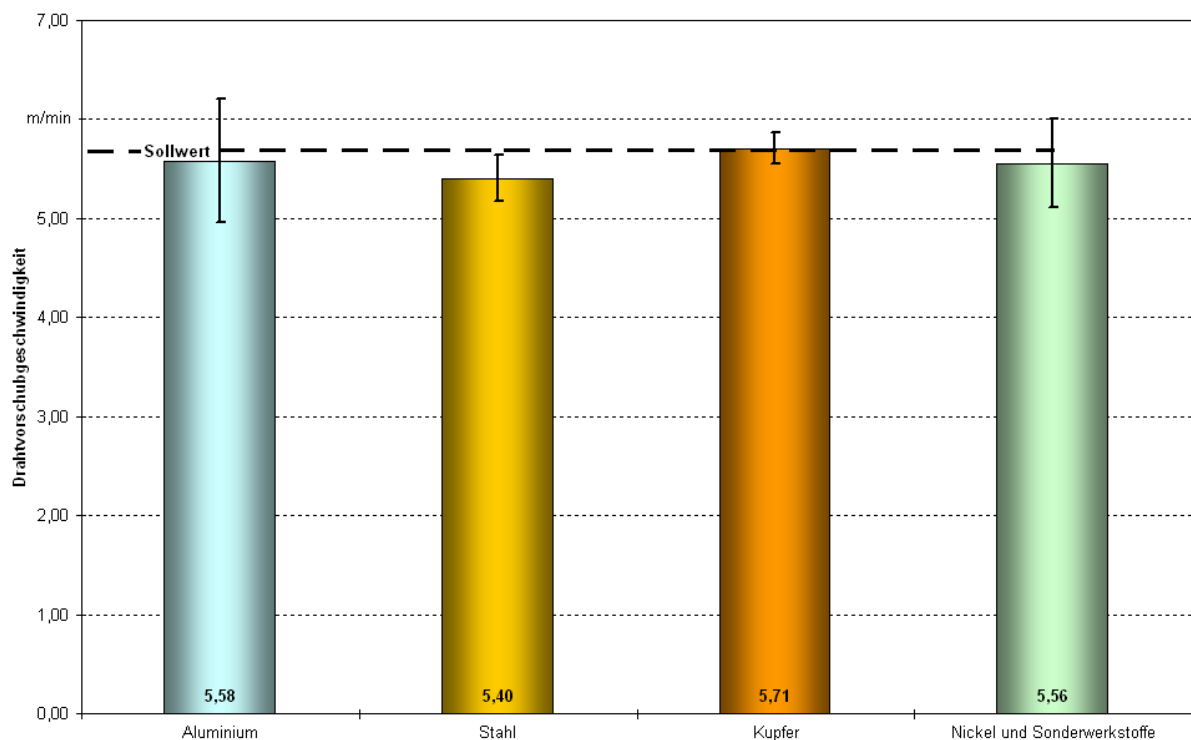


Abbildung 6.2: Ergebnisse der Versuche bei einem Sollwert von $5,69 \text{ m/min}$

Tabelle 6.4 spiegelt die Ergebnisse der Versuchsreihen mit dem Sollwert $8,53 \text{ m/min}$ wieder. Insgesamt wurden 574 Messungen durchgeführt, wobei 64 Bildkombinationen, wiederum vorwiegend bei Aluminium und Nickel/Sonderwerkstoffen, aussortiert wurden. Der maximale relative Fehler lag hier bei Stahl und Kupfer deutlich unter 5 %, wobei Kupferdrähte wiederum die besten Ergebnisse lieferten. Bei Aluminium und Nickel/Sonderlegierungen lag der relative Fehler etwas über 10 %. Durch Abbildung 6.3 wird der beschriebene Sachverhalt nochmals deutlicher wiedergegeben. Auch hier liegen die Geschwindigkeiten des Stahldrahtes ca. $0,21 \text{ m/min}$ unter dem Sollwert. Dies bestätigt die Vermutung des bereits oben beschriebenen Schlupfes.

Tabelle 6.4: Ergebnisse für den Sollwert $8,53 \text{ m/min}$

Werkstoff	Alu	Stahl	Kupfer	Ni- und Sonderw.
Istwerte im Mittel [m/min]	7,94	8,09	8,55	8,23
Max. absoluter Fehler [m/min]	1,05	0,23	0,19	1,11
Max. relativer Fehler [%]	12,33	2,74	2,28	13,01
Anzahl Messungen gesamt	123	123	123	205
Anzahl gültige Messungen	103	121	123	163

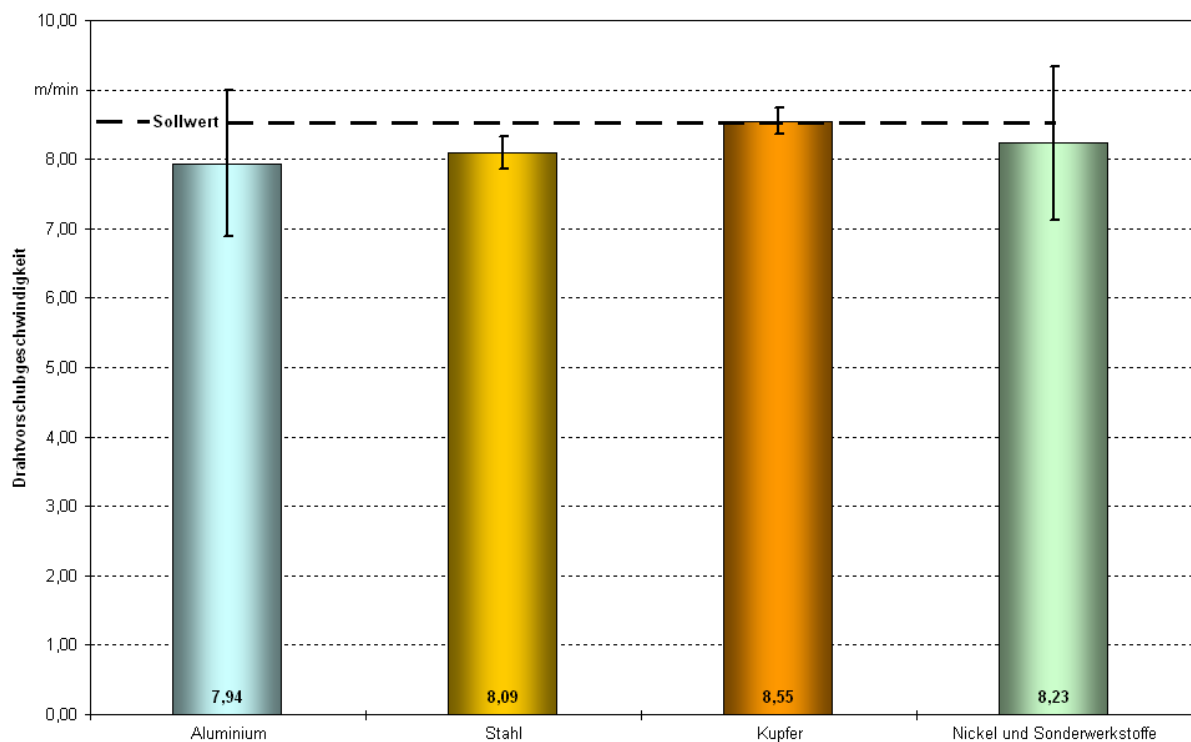


Abbildung 6.3: Ergebnisse der Versuche bei einem Sollwert von $8,53 \text{ m/min}$

7 Zusammenfassung und Ausblick

Die korrekte Einhaltung der Drahtvorschubgeschwindigkeit ist für die Qualität des Schweißergebnisses von entscheidender Bedeutung. Moderne Schweißenergiequellen besitzen, der aktuellen Entwicklung des Elektro- und Elektronikmarktes folgend, bereits Komponenten und Algorithmen, um alle elektrischen Schweißparameter reproduzierbar konstant zu halten sowie dieses auch zu protokollieren. Im Bereich der nichtelektrischen Parameter – Drahtvorschub und Gasdurchfluss – befindet man sich auf einem Entwicklungsstand von ca. 1970/1980. Dies zeigt die Notwendigkeit der in diesem Projekt durchgeführten Forschungsaktivitäten. Zum Einsatz kommen Kamerasysteme, welche mittlerweile Preisbereiche von unter 500 € erreicht haben. Obwohl die Kosten zunächst sekundär sind, spielen diese in der Gesamtbetrachtung für einen späteren Praxiseinsatz eine entscheidende Rolle.

Unter Nutzung modernster Verfahren der mathematischen Signalverarbeitung – der Wavelettransformation – wurden die anfallenden enormen Datenmengen dieses Messverfahrens verarbeitet und ausgewertet. Ziel dabei war die Erreichung der Echtzeitfähigkeit, wiederum unter dem Blickwinkel des späteren Praxiseinsatzes. Dieser Anspruch wurde durch die effektive Verarbeitung der Signale zum Ende der Projektlaufzeit erfüllt. Die durchgeführten Versuche an den verschiedensten Drahtqualitäten haben die Eignung des Kamerasystems zur berührungslosen Drahtvorschubgeschwindigkeitsmessung bewiesen. Diese Aussagen wurden mit nahezu 2300 Einzelmessungen belegt.

Darüber hinaus wurden zahlreiche Versuche durchgeführt, um das System zu kalibrieren und zu optimieren. So wurde mit verschiedenen Filtern gearbeitet, um das stark gebündelte Licht der Beleuchtungseinrichtungen in eine optimalere diffuse Beleuchtung umzuwandeln. Dieser Versuchsteil soll grundlegend im nächsten folgenden Abschnitt des Projektes untersucht werden und sei deshalb hier nur kurz erwähnt.

Der in den Versuchen maximal auftretende Fehler von ca. 13 %, welcher durch die hohe Anzahl der Versuche statistisch sehr gut abgesichert ist, verdeutlicht das Potenzial gegenüber konventionellen Drahtvorschubsystemen. Dazu ist anzumerken, dass als „richtiger Wert“ die Drehzahl des Antriebssystems zu Grunde gelegt wurde. Dies dürfte in erster Näherung auch korrekt sein, da die auftretenden Reibungen sehr gering sind. Auffällig ist allerdings, dass sehr weiche (Aluminium) und sehr harte Drähte (Nickel) höhere Abweichungen verursachen. Dies kann zu auftretenden Verformungen und verstärkten Schlupf und damit verbundenem „falschen Sollwert“ führen. D. h. das untersuchte Messsystem hat die genaue sich schnell ändernde Geschwindigkeit korrekt erfasst. Da im Moment, unseres Wissens nach, noch kein schnelleres Messsystem als das hier vorgestellte existiert, ist ein Nachweis dieser Vermutung schwer möglich. Denkbar wäre die Aufnahme der Drahtoberfläche mittels einer High-Speed-Kamera. Da solch ein System seit kurzer Zeit am Institut vorhanden ist, wäre ein diesbezüglicher Testdurchlauf in der nächsten Projektphase durchführbar. Der maximal auftretende Fehler von ca. 6 % (ca. 2,5 % minimal) bei den Kupferwerkstoffen untermauert diese Aussagen. Andererseits wäre die Oberflächenfarbe für die Qualität der Messung von Bedeutung. Die Oberflächenrauheit ist nachweislich nicht die Ursache der unterschiedlichen Auswertequalität.

Insbesondere die Aufbereitungs- und Auswertealgorithmen in Kombination mit den Algorithmen zur statistischen Absicherung der Ergebnisse lassen im Wesentlichen kaum noch Raum für weitere Optimierungen. Das größte Potenzial für Qualitätsverbesserungen liegt im Kamera- sowie Beleuchtungssystem. Auf Grund der positiven Ergebnisse sollte der Versuch unternommen werden, die Ergebnisse für eine Drahtvorschubgeschwindigkeitsregelung zu nutzen.

8 Literaturverzeichnis

- [1] Schneider, R.: Basiseigenschaften von Wavelets zur Multiskalenauswertung digitaler Signale, ZPF-Zeitschrift für Photogrammetrie und Fernerkundung 3 (1996), S. 75-83
- [2] Tsay, R.; Schneider, R; Wrobel, B.: Wavelet-gestützte Oberflächenrekonstruktion mit dem Verfahren des Facetten-Stereosehens, ZPF-Zeitschrift für Photogrammetrie und Fernerkundung 3 (1996), S. 83-100
- [3] Daubechies, I.: Ten Lectures on Wavelets. CBMS-NSF Regional Conf. Series in Applied Mathematics. Vol. 61, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992
- [4] Mallat, S.: A Wavelet Tour of Signal Processing. Second Edition Academic Press, 1999
- [5] Louis, A.K; Maaß, P; Rieder, A.: Wavelets. Teubner Studienbücher Mathematik, Teubner-Verlag Stuttgart, 1994
- [6] Dilthey, U; Reisgen, U: Untersuchungen an Drahtvorschubsystemen zum Metall-Schutzgasschweißen. Schweißen und Schneiden 47 (1995) 01, DVS Verlag, Düsseldorf, 1995
- [7] Matthes, K. - J.: Einsatz von optischen Sensoren für die Automatisierung des Lichtbogenschweißens. Bericht - TH Karl-Marx-Stadt Wissenschaftsbereich Fügetechnik und Montage, 1981
- [8] Matthes, K. - J.: Einsatz von Sensoren für das automatische Metallaktivgasschweißen. Wissenschaftliche Zeitschrift der TH Karl-Marx-Stadt, 27 (1985) 4, S. 542-549
- [9] Matthes, K. - J.; Schuricht, K.: Einsatz von low-cost-Messwertaufnehmer-Komponenten für die Überwachung des Metallschutzgasschweißens. DVS Berichte 146 (1992), S.70-74
- [10] Matthes, K.-J.; Kohler, T: Miniradarsensorik in der Schweißtechnik - Grundlagen und Stand der Technik. Schweißen und Schneiden 52 (2000) 10, DVS Verlag, Düsseldorf, 2000
- [11] Wang, Y., Becker, W.-J.: Trennung des Messabstandes von den Materialeigenschaften eines Messobjektes mittels eines Wirbelstromsensors. AMA-Kongressband Sensor 95 S.231-236, Nürnberg 1995
- [12] Engelberg, T.; Stein, D; Zeitler, R: Berührungslose Geschwindigkeitsmessung fester Oberflächen. tm 64 (1997) 10 S.380-386; Oldenbourg-Verlag
- [13] Gätke, J.: Untersuchungen zu akustischen Durchflussmessungen nach dem Laufzeitverfahren in Clamp-on-Anordnung bei gestörten Strömungsprofilen. Berichte aus der Elektrotechnik, Shaker-Verlag 2000
- [14] Kohlrusch, J. und Janocha H.: Einsatz eines modifizierten Kalmanfilters für das Laufzeit-Korrelationsverfahren. tm 61(1994) S. 33-39, Oldenbourg-Verlag

- [15] Wollnack, J.; Fenske, A.; Nicolaysen, E.; Rall, K.: Dichtheitskontrolle von Zapfverschlüssen in der Getränkeindustrie mit Methoden der digitalen Bildverarbeitung. tm 61 (1994) 2 S.51-59, Oldenbourg-Verlag
- [16] Fröhlich, H.; Gloning, J.; Lau, B.: Berührungslose optische Erfassung zweidimensionaler Translationsbewegungen. Sensor Magazin 1 (1995) S.8-12
- [17] Dietmayer, K.; Krüger, K.: Verfahren zur Berührungslosen Erfassung ebener Bewegungen. tm 61 (1994) 2 S.65-74, Oldenbourg-Verlag
- [18] Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press, 1998
- [19] de Vore, R. A.: Nonlinear Approximation. Acta Numerica (1998), S. 51-150
- [20] Mallat, S.: A Wavelet Tour of Signal Processing, Second Edition. Academic Press, 1999, S. 404-405
- [21] Griebel, M.; Oswald, P.; Schiekofer, T.: Sparse Grids for Boundary Integral Equations. Numerische Mathematik 83 (1999) 2, S.279-312, auch als SFB 256 Report 554, Universität Bonn <http://wissrech.iam.uni-bonn.de/research/pub/-griebel/integralgl.ps.gz>
- [22] Dahmen, W.; Prössdorf, S.; Schneider, R.: Multiscale methods for pseudo-differential equations on manifolds. Wavelet Analysis and its Applications 5 (1994) S. 385-424, Academic Press
- [23] Dahmen, W.; Schneider, R.; Xu, Y.: Nonlinear functionals of wavelet expansions - Adaptive reconstruction and fast evaluation. Numerische Mathematik 86 (2000), S. 49-101
- [24] Dahmen, W.; Kunoth, A.; Schneider, R.: Operator equations, multiscale concepts and complexity, Lectures in Applied Mathematics. American Mathematical Society 32 (1996), S. 225-261
- [25] Demant, C.; Streicher-Abel, B.; Waszkewitz, P.: Industrielle Bildverarbeitung, (1998), Springer Verlag
- [26] Jähne, B.: Digitale Bildverarbeitung, (1997), Springer Verlag
- [27] Weller, U.: Einsatz der digitalen Bildverarbeitung zur Bewegungsanalyse glühender Rohre im Walzprozess, (1997), Shaker Verlag
- [28] Joppich, M.: Schätzverfahren zur Genauigkeitssteigerung der Geschwindigkeitsmessung über Grund nach dem Dopplerprinzip, (1994), VDI-Verlag
- [29] Schmidt, U.: Digitale Videotechnik, (1996), Franzis-Verlag
- [30] Heck, A.: Farbcodes und Farbmodelle. <http://www.beta45.de/farbcodes/theorie/-farbmodelle.html>
- [31] Bourgeois, C.; Schneider, R.: Biorthogonal wavelets for the direct integral formulation of the heat equation. Preprint SFB 393 S. 00-14, TU Chemnitz 2000

- [32] Schneider, R.: Basiseigenschaften von Wavelets zur Multiskalenauswertung digitaler Signale, ZPF-Zeitschrift für Photogrammetrie und Fernerkundung 3 (1996), S. 75-83
- [33] Kohler, T.: Einsatz digitaler Algorithmen zur Basisregelung an Spritzgießmaschinen. Diplomarbeit, 1996
- [34] Graps, A.: An Introduction to Wavelets. IEEE Computational Science and Engineering Vol. 2, Num. 2, by IEEE Computer Society 1995
- [35] Hoffmann, H.: Ein Beitrag zur Analyse von hoch aufgelösten Elektrokardiogrammen. Diplomarbeit (1999), Brandenburgische Technische Universität Cottbus
- [36] Ngo, Minh-Ha: Ein Beitrag zur automatisierten Specklebild-Auswertung. VDI Fortschrittberichte 8 (1988) 149, VDI-Verlag Düsseldorf
- [37] Schröder, Hartmut: Mehrdimensionale Signalverarbeitung. Algorithmische Grundlagen für Bilder und Bildsequenzen (1998), Teubner-Verlag Stuttgart
- [38] Radinger, Christoph: Ein Beitrag zur Qualitätskontrolle bei Bandverarbeitungsanlagen mit einem selbst kalibrierenden Mehrkamerasystem. VDI Fortschritt-Berichte 20 (1996) 194, Rechnerunterstützte Verfahren, VDI-Verlag Düsseldorf
- [39] Schuster, Rolf: Objektverfolgung in Farbbildfolgen. Dissertationen zur künstlichen Intelligenz (1996) 115, Infix-Verlag Sankt Augustin
- [40] Dahlke, S.; Dahmen, W.; Hochmuth, R.; Schneider, R.: Stable Multiscale bases and local error estimation for elliptic problems. Applied Numerical Mathematics 23 (1997), S. 21-47
- [41] von Petersdorff, T.; Schneider, R.; Schwab, C.: Multiwavelets for second kind integral equations. SIAM Journal of Numerical Analysis, 34 (1997), S. 2212-2227
- [42] Schneider, R.; Levin, P.-L.; Spasojevic, M.: Multiscale compression of BEM equations of electrostatic systems. IEEE Transaction on Dielectrics and Electric Insulation, 3 (1996), S. 482-493.
- [43] Spasojevic, M.; Schneider, R.; Levin, P.-L.: On the creation of sparse boundary element matrices for two dimensional electrostatic problems using the orthogonal Haar wavelet. IEEE Transaction on Dielectrics and Electric Insulation, 4 (1997), S. 249-258
- [44] Konik, M.; Steidl, G.; Schneider, R.: Matrix compression by discrete wavelet transforms, Proceedings zur Konferenz Approximation Theory 8, (Eds. Shumaker et al), 1995.
- [45] Konik, M.; Steidl, G.; Schneider, R.: Matrix sparsification by discrete wavelet transforms. Proceedings des ICIAM 95 - Kongresses, 1995

- [46] Schneider, R.: Multiskalen- und Wavelet-Matrixkompression: Analysis basierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme. Habilitationsschrift, 1995, Advances in Numerical Mathematics, Teubner-Verlag Stuttgart, 1998
- [47] Dahmen, W.; Kleemann, B.; Prössdorf, S.; Schneider, R.: Multiscale methods for pseudodifferential equations. Proceedings des ICIAM 95 - Kongresses
- [48] Dahmen, W.; Kleemann, B.; Prössdorf, S.; Schneider, R.: Multiscale methods for the solution of the Helmholtz and Laplace equations. in Wendland, W.: Boundary Element Methods. Reports from the Final Conference of the Priority Research Programme 1989-1995 of the German Research Foundation, Stuttgart, Springer-Verlag, 1997
- [49] Kleemann, B.; Rathsfeld, A.; Schneider, R.: Multiscale methods for Boundary Integral Equations and their application to boundary value problems in scattering theory and geodesy. Boundary Element Methods-Algorithm, in Notes on Numerical Fluid Mechanics, 48 (1996), S. 1-44, Vieweg-Verlag
- [50] Dahmen, W.; Schneider, R.: Wavelets with complementary boundary conditions, Function spaces on the cube, Result. Math., 34 (1998), S. 255-293
- [51] Dahmen, W.; Schneider, R.: Composite wavelet bases for operator equation. Math. Comp., 68 (1999), S. 1533-1567
- [52] Levin, P.-L.; Spasojevic, M.; Schneider, R.: Creation of sparse boundary element matrices for 2-D and axi-symmetric electrostatic problems using the bi-orthogonal wavelets. Preprint SFB 393/97-29, erscheint in IEEE Transaction on Dielectrics and Electric Insulation
- [53] Konik, M.; Schneider, R.: Object-oriented implementation of multiscale methods for boundary integral equations. erscheint in Concepts of Numerical Software in Notes on Numerical Fluid Mechanics, Vieweg-Verlag
- [54] Dahmen, W.; Schneider, R.: Wavelets on manifolds I. Construction and domain decomposition. SIAM J. Mathematical Analysis, 31 (1999), S.184-230
- [55] Perez, C.; Paiva, F.; Harbrecht, H.; Schneider, R.: Biorthogonal wavelet approximation for the coupling of FEM-BEM. Preprint SFB393/99-32, (1999) erscheint Numerical Mathematics
- [56] Perez, C.; Schneider, R.: Wavelet Galerkin methods for boundary integral equations and the coupling with FEM. Preprint SFB393/99-33 (1999)
- [57] Dahmen, W.; Kunoth, A.; Schneider, R.: Wavelet least squares methods for boundary value problems. Preprint SFB393/99-34., (1999) eingereicht bei SIAM J. Numerical Analysis
- [58] Perez, C.; Paiva, F.; Harbrecht, H.; Schneider, R.: Multiscale preconditioning for the coupling of FEM-BEM. Preprint SFB393/00-07. (2000)

- [59] Rathsfeld, A.; Schneider, R.: On a quadrature algorithm for the piecewise linear collocation applied to boundary integral equations. SFB 393 Preprint 00-15, (2000)
- [60] Harbrecht, H.; Schneider, R.: Wavelet Galerkin Schemes for 2D BEM. SFB 393 Preprint 00-15, (2000); referiert und erscheint Proceedings of 11th TMP.
- [61] Louis, A.K.; Maaß, P.; Rieder, A.: Wavelets Theory and Applications. John Wiley & Sons, New York, 1997
- [62] Paeth, A. W.: "A Fast Algorithm for General Raster Rotation", Proceedings Graphics Interface '89, Canadian Information Processing Society (1986), S. 77-81
- [63] Abmayr, W.: Einführung in die digitale Bildverarbeitung. B.G. Teubner, Stuttgart, 1994
- [64] Haberäcker, P.: Praxis der Digitalen Bildverarbeitung und Mustererkennung. Carl Hanser Verlag, München, 1995
- [65] Jähne, B.: Digitale Bildverarbeitung. Springer, Berlin Heidelberg, 1996
- [66] Schneider, N.: Kanten hervorhebung und Kantenverfolgung in der industriellen Bildverarbeitung, Fortschritte der Robotik 6. Vieweg, Braunschweig Wiesbaden, 1990

9.1 Schaltplan zur synchronisierten Lichtansteuerung

9.2 BS2SX-Programmcode (Motorregler)

BS2SX-Programmcode (Motorregler); Teil 1

```
'{$STAMP BS2SX}
'{$PORT COM2}
'Motorregler-06_fuer_Maxon-Motor.bsx
```

'Lesen der Eingänge; INP_IGR_A, (sinnvolle Werte der low und High-Zeiten sind (25..100)*2us bei BS2)

'Besonderheiten:

- '-Anfangsrichtung immer konstant
- '-schnellere Berechnung ohne Unterprogramme
- '-Ausgabe der Geschwindigkeitsinformation
- '-Hinzufügen eines Reglerfreigabeausganges
- '-Überwachung der Endschalter und Stoppen der Reglerfreigabe
- '-über Flip-Flop-Schaltung Auswertung der Richtung und Übergabe an pin3
- '-DI für Richtungsinformation aus FF-Schaltung
- '-Division immer unter Vorzeichenbeachtung notwendig
- '-Vorgabe von Sollwertverläufen für die Geschwindigkeit (SIN, RAMPE, TRAPEZ)
- '-Test der Durchlaufzeiten und setzen der PID-Parameter
- '-Umschaltung nach Wegberechnung
- '-Rampen bei Richtungsumkehr
- '-verbesserte Berechnung fuer konstante Durchlaufzeit

```

'-----
DUMMYWORD    var word          'word-Hilfsvariable
DUMMYBYTE    var byte          'Byte-Hilfsvariable
DUMMYBIT     var bit           'Bit -Hilfsvariable

DEB           con 0             'debug ja oder nein
TA            con DEB*22+24     'Ta [ms] Abtastzeit, ohne Debug= 24ms, mit Debug= 40-60ms
'-----

'Strecke
IGR_A         var word          'Istwert (Pulsanzahl, Geschwindigkeit) IGR-Wert Kanal A
WEG           var word          'Istwert des Weges
W_KORR        var word          'Variable für Sollwertüberlagerung (W=W_0*W_KORR)
K_AMPLITUDE   con 5             'Divisor für Amplitude (1..255) 20 ist ganz gut
K_FREQUENZ    con 2             '[Hz] Sollfrequenz (max. 10 Hz)
ZAEHLER       var word          'Variable für Ändern der Sollwertkorrektur
ZAEHLRI       var bit           'Variable für Zaehrichtung pos.=0, neg.=1
ESV           var bit           'Endschalter vorn
ESH           var bit           'Endschalter hinten
RFRE          var bit           'Reglerfreigabe
DR_IST        var bit           'aktuelle Drehrichtung (0=links, 1=rechts)
INIT_BIT      var bit           'Bit zum 1-maligen Nullen des Weges
INIT_WEG      con 1000          'Weg für Initialisierung
DRU_NEG_WEG   con 0+INIT_WEG    'Umschaltwert für Weg
DRU_POS_WEG   con 57300+INIT_WEG 'Umschaltwert für Weg
BR_BS_WEG     con 360           'Brems- und Beschleunigungsweg
W_0           con 20            'Sollwert (-128 bis 127) fuer volle Fahrt, U=30 ca. 12.5 kHz
                                'IGR-Signal 115 = 10V -105 = -10V
KS            con 1             'stat. Streckenverst. KS_neu=1 (fuer hoehere Genauigkeit kleiner machen)
MZ            con 33            'Messzeit in 0,4ms (Linearachse MZ=10)
W             var word          'aktueller Sollwert
U             var word          'Ausgabewert (U_0)

'Regler:
E_0           var word          'e(k) Fehler
E_1           var word          'e(k-1) Fehler alt
E_2           var word          'e(k-2) Fehler alt
U_0           var word          'u(k) Stellgröße
U_1           var word          'u(k-1) Stellgröße alt

KDEB          con DEB*1         'Hilfskonstante wegen verbotener Klammer in CON
KKRIT         con 4             'Kkrit=4 bei W=15 (3 bei DEBUG)
TKRIT         con DEB*120+100   'Tkrit=100ms (220 ms bei DEBUG)
CF            con 50            'Korrekturfaktor fuer Erhoehung der Genauigkeit (max: 50)
```

BS2SX-Programmcode (Motorregler); Teil 2

```

'TV=0,12*Tkrit
TV          con 12*TKRIT/100          'Tv [ms] Vorhaltzeit (D-Anteil) (0 fuer Test)
'TN=0,5*Tkrit
TN          con TKRIT/2              'Tn [ms] Nachstellzeit (I-Anteil) (800 fuer Test)
'K=0,6*Kkrit
K           con 6*KKRIT/10          'K Verstärkung (P-Anteil) (1 fuer Test)
Q2          con 12*TKRIT/TA*CF/100   'immer positiv
'Q2         con TV*CF/TA             'immer positiv
Q1          con TA*CF*2/TKRIT-CF-Q2-Q2
'Q1         con TA*CF/TN-CF-Q2-Q2

Q0          con Q2+CF               'immer positiv

```

```

RICHTUNG_SOLL var bit
'++++++Portbelegung++++++
'Eingänge:
DI_IGR_A     con 0                  'Dateneingang für Frequenz IGR A
DI_IGR_A_PIN var in0               'Dateneingang für Frequenz IGR A
DI_ESV_PIN   con 1
DI_ESH_PIN   con 2
DI_ESV       var in1              'Eingang für Endschalter vorn
DI_ESH       var in2              'Eingang für Endschalter hinten
DI_DR        con 3                'Dateneingang für Drehrichtung
DI_DR_PIN    var in3              'Dateneingang für Drehrichtung

'Ausgänge:
DO_RFRE_PIN  con 4
DO_RFRE      var out4             'Ausgang für Reglerfreigabe BN6508
AO_PWM       con 5                'PWM-Ausgang
LIVE         con 6                'Wackelt, wenn alles IO
DO_DRU_PIN   con 7
DO_DRU       var out7             'Ausgang für Drehrichtungsumkehr
DO_V         var outh             'Geschwindigkeitsausgang

```

```

'-----Hauptprogramm-----
'Initialisierung nach Programmstart:
gosub INIT
DURCHLAUF:'Eingabesteuerung ist aktiv
gosub IOVERWALTUNG          'IOs lesen und setzen
'gosub FEHLERTEST           'derzeit nur Test, ob beide Endschalter an
'gosub SOLLWERT_KORR_SIN    'Sollwertkorrektur Sinus generieren
'gosub SOLLWERT_KORR_LIN    'Sollwertkorrektur Dreieck generieren
'gosub SOLLWERT_KORR_TRAP   'Sollwertkorrektur TRAPEZ generieren

'Fehler berechnen
'===== PID-Regler Algorithmus =====
'Sichern der alten Werte
E_2=E_1
E_1=E_0
E_0=(W+W_KORR)-IGR_A
U_1=U_0

if IGR_A.bit15=0 then ENDE_IGR_NEG
WEG=WEG-(-IGR_A*TA*10/MZ/4)
goto ENDE_IGR_POS
ENDE_IGR_NEG:
WEG=WEG+(IGR_A*TA*10/MZ/4)
ENDE_IGR_POS:

'debug cr, "WEG:", sdec WEG
'debug cr, "Q0:",sdec Q0," Q1:",sdec Q1," Q2:",sdec Q2

'Berechnung aktueller Wert
'U_0 = Q0*E_0, Q0 nie negativ
U_0=Q0*E_0

U=U-128
'debug cr, "W:",sdec W,"X:",sdec IGR_A,"E0:",sdec E_0,"U_1:",sdec U_1,"U:", sdec U

```

BS2SX-Programmcode (Motorregler); Teil 3

```

'U_0=U_0 + Q1*E_1, Q1 kann negativ sein!!
U_0=U_0+(Q1*E_1)

'debug ",U01:",sdec U_0

'U_0=U_0 + Q2*E_2, Q2 immer positiv
U_0=U_0+(Q2*E_2)

'debug ",U02: ",sdec U_0

'U_0=U_1 + U_0*K, K immer positiv
U_0=U_1+(U_0*K)

'fuer Schwingversuch reiner P-Regler
'U_0=E_0*K*CF

'debug ",U03: ",sdec U_0

'----- Begrenzung auf Byte-Variable -----
if U_0.bit15=0 then ENDE_STP_0
  U=-U_0/CF
  U=-(U max 128)
  goto ENDE_STN_0
  ENDE_STP_0:
  U=U_0/CF max 127
  ENDE_STN_0:
  'Stellwert negativ
  'Stellwert positiv

'U=W

'----- Drehrichtungsauswertung -----
'+++ umschalten wegen Endschalter +++
RFRE=0
if U.bit7=0 then ENDE_STP
RFRE=ESV^1
if ESV^1 then ENDE_DRUV
  W=W_0
  if INIT_BIT then ENDE_WEGINIT
  WEG=0
  INIT_BIT=1
  ENDE_WEGINIT:
  ENDE_DRUV
  goto ENDE_STN
  ENDE_STP:
  RFRE=ESH^1
  if ESH^1 then ENDE_DRUH
    W=-W_0
    ENDE_DRUH
    ENDE_STN:
    'Reglerfreigabe, wenn nicht ES vorn gekommen
    'Drehrichtungsumkehr vorn

'+++ umschalten wegen Weg +++
if WEG<DRU_POS_WEG then ENDE_POS_DRU
  W=-W_0
  ENDE_POS_DRU:
  if WEG>DRU_NEG_WEG then ENDE_NEG_DRU
    W=W_0
    ENDE_NEG_DRU:
    'Reglerfreigabe, wenn nicht ES hinten gekommen
    'Drehrichtungsumkehr hinten

'!!!!ACHTUNG!!! bei DEBUG: Schalter DEB oben auf 1 setzen
'debug cr,"X: ",sdec IGR_A," W:",sdec W," S:",dec WEG

'----- Normieren auf 0..255 -----
U=U+128
'U=U+128
'Stellwert auf 0..255 normieren)

goto DURCHLAUF

IOVERWALTUNG:
'Eingänge-----
DR_IST=DI_DR_PIN
count DI_IGR_A,MZ,IGR_A
IGR_A=IGR_A/KS 'Normierung auf KS_neu=1
'Drehrichtung
'zählt 0,4ms * MZ lang

```

BS2SX-Programmcode (Motorregler); Teil 4

```

if DR_IST=0 then ENDE_DR_NEG
  IGR_A=-IGR_A
ENDE_DR_NEG:
'debug cr,"IGR_A: ",sdec IGR_A," DR: ",dec DR_IST

ESV=DI_ESV
ESH=DI_ESH
'Ausgänge-----
DO_V=IGR_A
DO_RFRE=RFRE

pwm AO_PWM,U.lowbyte,10 '40 Zyklen von pwm bei 100/255
'Lebensbit:
toggle LIVE
return

INIT:
'Richtung setzen
input DI_IGR_A
input DI_ESV_PIN
input DI_ESH_PIN
output AO_PWM
output LIVE
output DO_DRU_PIN
output DO_RFRE_PIN
dirh=%11111111 'alles Ausgaenge
high LIVE
low DO_RFRE
'Variablen vorbesetzen
U_0=0
U_1=0
E_0=0
E_1=0
E_2=0
RICHTUNG_SOLL=1
INIT_BIT=0
WEG=INIT_WEG
W_KORR=0
W=W_0+W_KORR
ZAEHLER=0
ZAEHLRI=0
return

FEHLERTEST:
'wenn beide Eingänge aktiv, dann Fehler
if (ESV & ESH)=0 then ENDE_FDR
  DUMMYBYTE=0
ENDE_FDR:
return

SOLLWERT_KORR_SIN:
ZAEHLER=ZAEHLER+(256*TA*K_FREQUENZ/10)
if ZAEHLER<25599 then ENDE_BSIN
  ZAEHLER=0
ENDE_BSIN:
W_KORR=sin(ZAEHLER/100)
'debug cr, "WK:",hex W_KORR
if W_KORR.bit15=0 then ENDE_W_K_NEG_SIN
  W_KORR=(-W_KORR/K_AMPLITUDE)
  goto ENDE_W_K_POS_SIN
ENDE_W_K_NEG_SIN:
  W_KORR=W_KORR/K_AMPLITUDE
ENDE_W_K_POS_SIN:
'debug cr, "x=",dec ZAEHLER, " y=", sdec W_KORR
return

SOLLWERT_KORR_LIN:
DUMMYWORD=256*TA*K_FREQUENZ/5

```

BS2SX-Programmcode (Motorregler); Teil 5

```
if ZAEHLER<25599 then ENDE_BLIN_HOCH
ZAEHLRI=1
ENDE_BLIN_HOCH:
if ZAEHLER>1 then ENDE_BLIN_RUNTER
ZAEHLRI=0
ENDE_BLIN_RUNTER:

if ZAEHLRI=0 then ENDE_ZAEHLPOS
ZAEHLER=ZAEHLER-DUMMYWORD
goto ENDE_ZAEHLNEG
ENDE_ZAEHLPOS:
ZAEHLER=ZAEHLER+DUMMYWORD
ENDE_ZAEHLNEG:

W_KORR=ZAEHLER/100-128

if W_KORR.bit15=0 then ENDE_W_K_NEG_LIN
W_KORR=(-W_KORR/K_AMPLITUDE)
goto ENDE_W_K_POS_LIN
ENDE_W_K_NEG_LIN:
W_KORR=W_KORR/K_AMPLITUDE
ENDE_W_K_POS_LIN:
return

SOLLWERT_KORR_TRAP:
ZAEHLER=ZAEHLER+(256*TA*K_FREQUENZ/10)

if ZAEHLER<25599 then ENDE_BTRAP
ZAEHLER=0
ENDE_BTRAP:

W_KORR=ZAEHLER/100-128

if W_KORR.bit15=0 then ENDE_W_K_NEG_TRAP
W_KORR=(-128/K_AMPLITUDE)
goto ENDE_W_K_POS_TRAP
ENDE_W_K_NEG_TRAP:
W_KORR=128/K_AMPLITUDE
ENDE_W_K_POS_TRAP:

'debug cr, "ZA:",sdec ZAEHLER,"WK:",sdec W_KORR
return
```


9.4 Matlab – Programmcode (Hauptbearbeitungsprozedur)

```
function [c,b]=processN(frame,sp,ep,LL,flen,luft,lv,flag,col);

sx=size(frame,2);
sy=size(frame,1);

hold on;
tic;
for j=sp:ep
    if flag
        subplot(2,1,1);
        hold off;
    end;
    % Um die Bewegungsrichtung zu ändern, wählt man eine aus den folgenden Zeilen
    [c(j-sp+1),b(j-sp+1)]=findshift(frame(1:2:sy-1,:j),frame(2:2:sy ,:,j),flen,luft,lv,LL,flag);
    % [c(j-sp+1),b(j-sp+1)]=findshift(frame(2:2:sy ,:,j),frame(1:2:sy-1,:j),flen,luft,lv,LL,flag);
    title(j);
    if flag
        subplot(2,1,2);
        hold on;
    end;
    plot(j-sp,c(j-sp+1),col);
    pause(0);
end;
totalTime=toc
TimePerFrame=totalTime/(ep-sp+1)
FPS=(ep-sp+1)/totalTime
%%%%%%%%%% end of the function processN
```

9.5 Matlab – Programmcode (Berechnung der Verschiebung)

```

function [s, Breite]=findshift(picA,picB,flen,luft,lv,LL,flag);
%output:
%s      - calculated shift
%Breite - calculted width of the wire
%
%input:
%picA - first frame(8-bit image!)
%picB - second frame(8-bit image!)
%flen - the length of the window
%luft - radius of the vertical region where the window moves,
%      should be grater than the skew of the wire
%lv   - the shift where the window from second frame should be taken
%LL   - level of downsampling, can be: 1, 2, 4, 8.
%flag - Show interim results
%
%Existing warnings:
%When the wire is detected incorrectly, there are following warnings:
%Wrong Detection of the wire: top boundary
%Wrong Detection of the wire: bottom boundary
%In this case parameters for "givebound" should be changed (increased)

if LL==8
    [s, Breite]=findshift8(picA,picB,flen,luft,lv,flag);
else
    if LL==4
        [s, Breite]=findshift4(picA,picB,flen,luft,lv,flag);
    else
        if LL==2
            [s, Breite]=findshift2(picA,picB,flen,luft,lv,flag);
        else
            if LL==1
                [s, Breite]=findshift1(picA,picB,flen,luft,lv,flag);
            else
                Error('Wrong downsampling level.');
            end;
        end;
    end;
end;
end;
%%%%%% end of the function findshift

```


Matlab – Programmcode (Berechnung der Verschiebung); Prozedur zur Verschiebungsberechnung ohne Beschleunigung

```
function [s, Breite]=findshift1(picA,picB,flen,luft,lv,flag)
picx=size(picA,2);
picy=size(picA,1);
% Search for bounds
[Up,Down]=GiveBound(picB(:,picx+1-flen-lv:picx-lv),picA(:,picx+1-flen-lv:picx-lv),10,0.2);

Breite=Down-Up;
bk=ceil(Breite/8);

y0=Up-bk;
y1=Down+bk;

%ähnlichkeitsfunktion
r=alike(picA,y0-luft,y1+luft,picB(y0:y1,picx+1-flen-lv:picx-lv),1,1);
ss=findmin(r);
s=ss-lv-1;
if flag
    mondraw(r);
    %ss=findmin(r);
    hold on;
    plot(ss,r(ss),'o');
    pause(0);
end;
%% % % % % % end of the function findshift1
```

Matlab – Programmcode (Berechnung der Verschiebung); Prozedur zur Verschiebungsberechnung mit Beschleunigung durch Mittelwert (Wavelet)-Transformation bis zum Niveau 2

```
function [s, Breite]=findshift2(picA,picB,flen,luft,lv,flag)
picx=size(picA,2);
picy=size(picA,1);
[Up,Down]= GiveBound(picB(:,picx+1-flen-lv:picx-lv),picA(:,picx+1-flen-lv:picx-lv),10,0.2);

Breite=Down-Up;
bk=ceil(Breite/8);

y0=Up-bk;
y1=Down+bk;

picA2=fhUI8(picA(y0-luft:y1+luft,:),picx/2);
picB2=fhUI8(picB(y0:y1,picx+1-flen-lv:picx-lv),flen/2);

r2=alike(picA2(:,1:picx/2),0 .y1+2*luft-y0,picB2(:,1:flen/2),3,1);

r=findmin(r2);
if flag
    mondraw(r2);
    hold on;
    plot(r,r2(r),'o');
    pause(0);
end;

s=2*(r-1);
if s<2
    s=2;
end;
if picx-flen-2<s
    s=picx-flen-2;
end;

r1=alike(picA(:,picx-flen+1-s-2:picx-s+2),y0-luft,y1+luft,picB(y0:y1,picx+1-flen-lv:picx-lv),3,1);

[a,b]=min(r1);
s=s+b-3-lv;

%% % % % % % end of the function findshift2
```

Matlab – Programmcode (Berechnung der Verschiebung); Prozedur zur Verschiebungsberechnung mit Beschleunigung durch Mittelwert (Wavelet)-Transformation bis zum Niveau 4

```

function [s,Breite]=findshift4(picA,picB,flen,luft,lv,flag);

picx=size(picA,2);
picy=size(picA,1);

[Up,Down]= GiveBound(picB(:,picx+1-flen-lv:picx-lv),picA(:,picx+1-flen:picx),10,0.2);

Breite=Down-Up;
bk=ceil(Breite/8);

y0=Up-bk;
y1=Down+bk;

picA2=fhUI8(picA(y0-luft:y1+luft,:),picx/2);
picA4=fhI16(picA2(:,1:picx/2),picx/4);

picB2=fhUI8(picB(y0:y1,picx+1-flen-lv:picx-lv),flen/2);
picB4=fhI16(picB2(:,1:flen/2),flen/4);

r4=alike(picA4(:,1:picx/4),0 ,y1+2*luft-y0,picB4(:,1:flen/4),3,1);

r=findmin(r4);
if flag
    plot(r4);
    ss=lm(r4);
    hold on;
    plot(ss,r4(ss),'*');
    plot(r,r4(r),'o');
    pause(0);
end;

s=2*(r-1);
if s<2
    s=2;
end;
if picx/2-flen/2-2<s
    s=picx/2-flen/2-2;
end;

r2=alike(picA2(:,picx/2-flen/2+1-s-2:picx/2-s+2),0 ,y1+2*luft-y0,picB2(:,1:flen/2),3,1);

[a,b]=min(r2);
s=2*(s+b-3);
if s<2
    s=2;
end;
if picx-flen-2<s
    s=picx-flen-2;
end;

r1=alike(picA(:,picx-flen+1-s-2:picx-s+2),y0-luft,y1+luft,picB(y0:y1,picx+1-flen-lv:picx-lv),3,1);

[a,b]=min(r1);
s=s+b-3-lv;

%the end
%%%%%%%% end of the function findshift4

```

Matlab – Programmcode (Berechnung der Verschiebung); Prozedur zur Verschiebungsberechnung mit Beschleunigung durch Mittelwert (Wavelet)-Transformation bis zum Niveau 8; Teil 1

```

function [s,Breite]=findshift8(picA,picB,flen,luft,lv,flag)
%output:
%s    - calculated shift
%Breite - calculted wideness
%
%input:
%picA - first frame(8-bit image!)
%picB - second frame(8-bit image!)
%flen - the length of the window
%luft - radius of the vertical region where the window moves,
%       should be grater than the skew of the wire
%lv    - the shift where the window from second frame should be taken
%flag - Show interim results
%
%Existing warnings:
%When the wire is detected incorrectly, there are following warnings:
%Wrong Detection of the wire: top boundary
%Wrong Detection of the wire: bottom boundary
%
%In this case parameters for "givebound" should be changed(increased)
%

picx=size(picA,2);
picy=size(picA,1);

[Up,Down]=GiveBound(picB(:,picx+1-flen-lv:picx-lv),picA(:,picx+1-flen:picx),10,0.2);

Breite=Down-Up;
bk=ceil(Breite/8);

y0=Up-bk;
y1=Down+bk;

if y0<luft+1
    y0=2*luft+1;
    Warning('Wrong Detection of the wire: top boundary. ');
end;
if y1>picy-luft
    y1=picy-2*luft;
    Warning('Wrong Detection of the wire: bottom boundary. ');
end;

picA2=fhUI8(picA(y0-luft:y1+luft,:),picx/2);
picA4=fhI16(picA2(:,1:picx/2),picx/4);
picA8=fhI16(picA4(:,1:picx/4),picx/8);

picB2=fhUI8(picB(y0:y1,picx+1-flen-lv:picx-lv),flen/2);
picB4=fhI16(picB2(:,1:flen/2),flen/4);
picB8=fhI16(picB4(:,1:flen/4),flen/8);

r8=alike(picA8(:,1:picx/8),0 ,y1+2*luft-y0,picB8(:,1:flen/8),1,1);

r=findmin(r8);
if flag
    mondraw(r8);
    hold on;
    plot(r,r8(r),'o');
    pause(0);
end;

s=2*(r-1);
if s<2
    s=2;
end;

```

Matlab – Programmcode (Berechnung der Verschiebung); Prozedur zur Verschiebungsberechnung mit Beschleunigung durch Mittelwert (Wavelet)-Transformation bis zum Niveau 8; Teil 2

```

if picx/4-flen/4-2<s
    s=picx/4-flen/4-2;
end;

r4=alike(picA4(:,picx/4-flen/4+1-s-2:picx/4-s+2),0 ,y1+2*luft-y0,picB4(:,1:flen/4),3,1);
[a,b]=min(r4);
s=2*(s+b-3);
if s<2
    s=2;
end;
if picx/2-flen/2-2<s
    s=picx/2-flen/2-2;
end;

r2=alike(picA2(:,picx/2-flen/2+1-s-2:picx/2-s+2),0 ,y1+2*luft-y0,picB2(:,1:flen/2),3,1);

[a,b]=min(r2);
s=2*(s+b-3);

if s<2
    s=2;
end;
if picx-flen-1<s
    s=picx-flen-2;
end;

r1=alike(picA(:,picx-flen+1-s-2:picx-s+2),y0-luft,y1+luft,picB(y0:y1,picx+1-flen-lv:picx-lv),3,1);

[a,b]=min(r1);
s=s+b-3-lv;

% % % % % end of the function findshift8

```

9.6 Matlab – Programmcode (Verschiebung mit Neigungskorrektur)

Prozedur zum Finden der Neigung

```

function b=giveSlope(picA,y0L,y1L,flen,ss)

picx=size(picA,2);
win=picA(y0L:y1L,1:flen);
for s=-ss:ss
    r(s+1:ss)=sum(sum(abs(win-picA(y0L+s:y1L+s,picx+1-flen:picx)))));
end;
[a,b]=min(r);
bb=findmin(r);
if bb~b
    warning('Function is not unimodular, selected local min is not equal to the global one');
    b=bb;
    %stop; %there is no statistic control here
end;
b=b-1-ss;
% % % % % end of the function

```

Verschiebeprozedur; Teil 1

```

function c=findshift(picA,picB,k,flen,LL);

picx=size(picA,2);
picy=size(picA,1);
pq=6;
pl=0.2;
[Up,Down] =GiveBound(picB(:,picx+1-flen:picx),picA(:,picx+1-flen:picx),pq,pl);
[UpL,DownL]= GiveBound(picB(:,1:flen),picA(:,1:flen),pq,pl);

```

Matlab – Programmcode (Verschiebung mit Rotation); Verschiebeprozedur; Teil 2

```

Breite=Down-Up;
BreiteL=DownL-UpL;
Breite=max(Breite,BreiteL);
bk=floor(Breite*0.5)+1;%cn
s1=floor((Down+Up)/2);
s2=floor((DownL+UpL)/2);
y0=s1-bk;
y1=s1+bk;
y0L=s2-bk;
y1L=s2+bk;

p=10;
skewA=giveSkew(picA,y0L,y1L,flen,p);
skewB=giveSkew(picB,y0L,y1L,flen,p);

if skewA
    picAr=rotator(picA,atan2(skewA,picx));
else
    picAr=picA;
end;
if skewB
    picBr=rotator(picB,atan2(skewB,picx));
else
    picBr=picB;
end;

lv=24;
luft=20;

r=alike(picAr,y0-luft,y1+luft,picBr(y0:y1,picx+1-flen-lv:picx-lv),1,3,1);
ss=findmin(r);
s=ss-lv-1;
% % % % end of the function

```

Rotationsprozedur 1

```

function picAr=rotator(picA,angle)

picx=size(picA,2);
picy=size(picA,1);

picAr=rotate(picA,angle);

pX1r=size(picAr,1);
pX2r=size(picAr,2);

s1=ceil((pX1r-picy)/2);
s1e=pX1r-picy-s1;
s2=ceil((pX2r-picx)/2);
s2e=pX2r-picx-s2;

picAr=picAr(s1:pX1r-s1e-1,s2:pX2r-s2e-1);
% % % % end of the function

```

Rotationsprozedur 2

```

function picAr=rotator(picA,angle)

picx=size(picA,2);
picy=size(picA,1);
picAr=yshear(picA,angle);

pX1r=size(picAr,1);

s1=ceil((pX1r-picy)/2);
s1e=pX1r-picy-s1;

picAr=picAr(1:pX1r-s1e-1,:);
% % % % end of the function

```

9.7 Matlab – Programmcode (Bestimmung der Kanten)

```
function [Up,Down]=GiveBound(pic1, pic2,s,lev);
%[Up,Down]=GiveBound(pic1, pic2,s,lev);
%Calculates the up and down side of moving wire.
% s - parameter of integration(middeling)
% lev - level of the noise.

ys=size(pic1,1);
x=sum(abs(pic1-pic2));
for i=s+1:ys-s
    y(i-s)=sum(x(i-s:i+s));
end;
v=minmax(y);
xx=v(1)+(v(2)-v(1))*lev;
k=1;
while y(k)<=xx
    k=k+1;
end;
Up=k+s;
k=size(y,2);
while y(k)<=xx
    k=k-1;
end;
Down=k+s;
%%%%%%%% end of the function
```

9.8 Matlab – Programmcode (Ähnlichkeitsfunktionen)

Variante 1

```
function r=alike(picA,Up,Down,luft,draht)
picx=size(picA,2);
ry=zeros(1,2*luft+1);
for y=-luft:luft
    ry(y+luft+1)=sum(sum(abs(picA(Up+y:Down+y,:)-draht))));
end;
[a,b]=min(ry);
hvers=b-1-luft;

r=zeros(1,picx);
r(1)=a;
for i=2:picx
    rv(1)= sum(sum(abs(picA(Up+hvers-1:Down+hvers-1,1:picx-i+1)-draht(:,i:picx))));
    rv(2)= sum(sum(abs(picA(Up+hvers :Down+hvers ,1:picx-i+1)-draht(:,i:picx))));
    rv(3)= sum(sum(abs(picA(Up+hvers+1:Down+hvers+1,1:picx-i+1)-draht(:,i:picx))));
    [a,b]=min(rv);
    r(i)=a;
    hvers=hvers+b-2;
end;
%%%%%%%% end of the function
```

Variante 2; Teil 1

```
function r=alike(pic,y0,y1,win,p,k,drflag);
%The function finds the norm of the difference
%between pictures win and pic for different shifts
%Picture win consists of an object to be found in picture pic
%
%In this version the warnings
%'Warning: Losing the trace of the wire - underflow'
%'Warning: Losing the trace of the wire - overflow'
%mean that the size of the window is incorrect,
%it should be bigger.
if nargin < 7, drflag=0; end;
```

Matlab – Programmcode (Ähnlichkeitsfunktionen), Variante 2; Teil 2

```

winx=size(win,2);
winy=size(win,1);

picx=size(pic,2);
picy=size(pic,1);

kx=picx-winx+1;
%ky=picy-winy+1;
ky=y1-y0+1-winy+1;

for i=1:ky
    ry(i)=sum(sum(abs(pic(y0-1+i:y0-1+winy+i-1, kx:picx)-win).^p));
end;
[a,b]=min(ry);
hvers=b-1;
r(1)=a;
if drflag
    gr(kx)=hvers;
end;

for i=2:kx
    for j=1:k
        rv(k-j+1) = sum(sum(abs(pic(y0-1+1+hvers-j:y0-1+winy+hvers-j,kx-i+1:picx-i+1)-win).^p));
        rv(k+j+1) = sum(sum(abs(pic(y0-1+1+hvers+j:y0-1+winy+hvers+j,kx-i+1:picx-i+1)-win).^p));
    end;
    rv(k+1) = sum(sum(abs(pic(y0-1+1+hvers :y0-1+winy+hvers, kx-i+1:picx-i+1)-win).^p));
    [a,b]=min(rv);
    r(i)=a;
    for lii=1:2*k+1
        if lii~=b
            if rv(lii)==a
                if abs(lii-k-1)<abs(b-k-1)
                    b=lii;
                end;
                warning('Trace is constant');
            end;
        end;
    end;
    hvers=hvers+b-k-1;
    if drflag
        gr(kx+1-i)=hvers;
    end;
    if hvers<1+k-y0
        hvers=1+k-y0;
        warning('Losing the trace of the wire - underflow');
    end;
    if hvers>picy-winy-y0+1-k
        hvers=winy;
        warning('Losing the trace of the wire - overflow');
    end;
end;
if drflag==1
    plot((y0+y1)/2+gr,'r*');
    pause(0);
end;
if drflag==2
    plot(winx:picx,(y0+y1)/2-gr(size(gr,2):-1:1),'r*');
    pause(0);
end;
% % % % % end of the function

```

9.9 Matlab – Programmcode (Lokale Minima)

Prozedur Variante 1

```
function r=findmin(f);

sz=size(f,2);
FirstDer =f(2:sz)-f(1:sz-1);

r=1;
sd=0;

for x=2:sz-1
    if ((FirstDer(x-1)<0)&(FirstDer(x)>0))
        SecondDer=FirstDer(x)-FirstDer(x-1);
        if SecondDer>sd
            r=x;
            sd=SecondDer;
        end;
    end;
end;
%% % % % % % end of the function

function r=findmin(f);
sz=size(f,2);
FirstDer =f(2:sz)-f(1:sz-1);

r=1;
sd=0;

for x=3:sz-2
    if ((FirstDer(x-1)<0)&(FirstDer(x)>0))
        SecondDer=-FirstDer(x-2)+16*FirstDer(x-1)-30*FirstDer(x)+16*FirstDer(x+1)-FirstDer(x+2);
        if SecondDer>sd
            r=x;
            sd=SecondDer;
        end;
    end;
end;
%% % % % % % end of the function
```

Prozedur Variante 2; Teil 1

```
function r=findmin(x);

n=size(x,2);
r=1;
if(n~=1)
    pn=1;
    i=1;
    if x(i)>x(i+1)
        inc=-1;
    else
        inc= 1;
    end;
    i=i+1;

    wwp=0;
    wwc=0;
    ww=-1;
    r=n;
    while i<n
        if inc== -1
            if x(i)<x(i+1)
                inc=1;
                wwc=x(pn)-x(i);
                pn=i;
            end;
        else
```


Matlab – Programmcode (Lokale Minima); Prozedur Variante 2; Teil 2

```

    if x(i)>x(i+1)
        inc=-1;
        wwp=x(i)-x(pn);
        if ww<min(wwc,wwp)
            ww=min(wwc,wwp);
            r=pn;
        end;
        pn=i;
    end;
    end;
    i=i+1;
end;
if inc==1
    wwc=x(pn)-x(n);
else
    wwp=x(n)-x(pn);
    if ww<min(wwc,wwp)
        ww=min(wwc,wwp);
        r=pn;
    end;
end;
end;
end;
%%%%%% end of the function

```

9.10 C++ – Programmcode (Rotation)

Im folgenden Programmcode sind eigentlich zwei Programme enthalten. Das erste dreht das Bild mittels ebener Verschiebung, das andere führt nur die lineare vertikale Verschiebung mit Mittelwertbildung durch. Um eine bestimmte Version auszuwählen, setzt man den dementsprechenden Schalter vor der Compilierung.

C++ – Programmcode (Rotation); Teil 1

```

#include "stdafx.h"
#include <math.h>
#include "bild.h"
#include <afxwin.h>

void HorizSkew(BYTE* pSrc, int Swidth,int Sheight,BYTE* pDst, int Dwidth,int Dheight, UINT uRow,int iOffset,double dWeight)
{
    int i;
    /*for (int i = 0; i < iOffset; i++)
        pDst[uRow+i*Dheight]=0;*/

    BYTE pxlOldLeft = 0;
    for (i = 0; i < Swidth; i++)
    {
        // Calculate weights
        BYTE pxlSrc =pSrc[uRow+i*Sheight];
        BYTE pxlLeft = BYTE ( double (pxlSrc) * dWeight );
        pxlSrc = pxlSrc - ( pxlLeft - pxlOldLeft);

        // Check boundaries
        if ((i + iOffset >= 0) && (i + iOffset < Dwidth))
            pDst[uRow+(i+iOffset)*Dheight]=pxlSrc;

        pxlOldLeft = pxlLeft;
    }
    // Go to rightmost point of skew
    i += iOffset;
    if (i < Dwidth)
        pDst[uRow+i*Dheight]=pxlOldLeft;

    /*while (++i < Dwidth)
        pDst[uRow+i*Dheight]=0;*/
} // CRotateByShear::HorizSkew

```

C++ – Programmcode (Rotation); Teil 2

```

void VertSkew(BYTE* pSrc, int Swidth,int Sheight,BYTE* pDst, int Dwidth,int Dheight, UINT uCol,int iOffset,double dWeight)
{
    int i;
    /*for (int i = 0; i < iOffset; i++)
        pDst[i+uCol*Dheight]=0;*/

    BYTE pxlOldLeft = 0;

    int iYPos;

    for (i = 0; i < Sheight; i++)
    {
        BYTE pxlSrc = pSrc[i+uCol*Sheight];

        iYPos = i + iOffset;

        BYTE pxlLeft =BYTE ( double (pxlSrc) * dWeight );
        pxlSrc =pxlSrc - ( pxlLeft - pxlOldLeft );

        // Check boundaries
        if ((iYPos >= 0) && (iYPos < Dheight))
            pDst[iYPos+uCol*Dheight]=pxlSrc;

        pxlOldLeft = pxlLeft;
    }
    // Go to bottom point of skew
    i = iYPos;
    if (i < Dheight)
        pDst[i+uCol*Dheight]=pxlOldLeft;

    /*while (++i < Dheight)
        pDst[i+uCol*Dheight]=0;*/
} // CRotateByShear::VertSkew

BYTE* Rotate45(BYTE* pSrc, int Swidth,int Sheight,SIZE *psDst, double dRadAngle)
{
    //double dRadAngle = dAngle * 3.14 / double(180); // Angle in radians
    double dSinE = sin (dRadAngle);
    double dTan = tan (dRadAngle / 2.0);

    // Calc first shear (horizontal) destination image dimensions
    SIZE sDst1;
    sDst1.cx = Swidth + int(double(Sheight) * fabs(dTan));
    sDst1.cy = Sheight;

    /****** Perform 1st shear (horizontal) *****/

    // Allocate image for 1st shear
    //CPxl *pDst1 = CreateNewBitmap (sDst1);
    BYTE *pDst1 = new BYTE[sDst1.cx*sDst1.cy];
    for (UINT u = 0; u < UINT(sDst1.cy); u++)
    {
        double dShear;

        if (dTan >= 0.0)
        {
            // Positive angle
            dShear = (double(u) + 0.5) * dTan;
        }
        else
        {
            // Negative angle
            dShear = (double (int(u) - sDst1.cy) + 0.5) * dTan;
        }
        int iShear = int (floor (dShear));

        HorizSkew(pSrc, Swidth,Sheight,pDst1, sDst1.cx,sDst1.cy,
            u,
            iShear,
            dShear - double(iShear));
    }
}

```

C++ – Programmcode (Rotation); Teil 3

```

/***** Perform 2nd shear (vertical) *****/

// Calc 2nd shear (vertical) destination image dimensions
SIZE sDst2;
sDst2.cx = sDst1.cx;
sDst2.cy = UINT (double (Swidth) * fabs (dSinE) + double (Sheight) * cos (dRadAngle)) + 1;
// Allocate image for 2nd shear
//CPxl *pDst2 = CreateNewBitmap (sDst2);
        BYTE *pDst2 =new BYTE[sDst2.cx*sDst2.cy];

double dOffset; // Variable skew offset
if (dSinE > 0.0)

{
    // Positive angle
    dOffset = double (Swidth - 1) * dSinE;
}
else
{
    // Negative angle
    dOffset = -dSinE * double (Swidth - sDst2.cx);
}

for (u = 0; u < UINT(sDst2.cx); u++, dOffset -= dSinE)
{
    int iShear = int (floor (dOffset));
                                VertSkew(pDst1, sDst1.cx,sDst1.cy,pDst2, sDst2.cx,sDst2.cy,
        u,
        iShear,
        dOffset - double(iShear));
}

/***** Perform 3rd shear (horizontal) *****/
        delete pDst1;

// Calc 3rd shear (horizontal) destination image dimensions
(*psDst).cx = UINT (double(Sheight) * fabs (dSinE) + double(Swidth) * cos (dRadAngle)) + 1;
(*psDst).cy = sDst2.cy;

// Allocate image for 3rd shear
//CPxl *pDst3 = CreateNewBitmap (*psDst);
        BYTE *pDst3 =new BYTE[(psDst).cx *(psDst).cy];

if (dSinE >= 0.0)
{
    // Positive angle
    dOffset = double(Swidth- 1) * dSinE * -dTan;
}
else
{
    // Negative angle
    dOffset = dTan * (double(Swidth - 1) * -dSinE + double(1 - (*psDst).cy));
}
for (u = 0; u < UINT((psDst).cy); u++, dOffset += dTan)
{
    int iShear = int (floor(dOffset));
    HorizSkew ( pDst2,sDst2.cx,sDst2.cy,pDst3,(psDst).cx,(psDst).cy,
        u,
        iShear,
        dOffset - double (iShear));
}
// Free result of 2nd shear
        delete pDst2;
// Return result of 3rd shear
return pDst3;
} // Rotate45

```

C++ – Programmcode (Rotation); Teil 4

```

BYTE* VertShear(BYTE* pSrc, int Swidth,int Sheight,SIZE *psDst, double dRadAngle)
{
    double dSinE = sin (dRadAngle);

    (*psDst).cx = Swidth;
    (*psDst).cy = UINT (double (Swidth) * fabs (dSinE) + double (Sheight) * cos (dRadAngle)) + 1;
    BYTE *pDst2 =new BYTE[( *psDst).cx*( *psDst).cy];

    double dOffset;    // Variable skew offset
    if (dSinE > 0.0)
    {
        // Positive angle
        dOffset = double (Swidth - 1) * dSinE;
    }
    else
    {
        // Negative angle
        dOffset = -dSinE * double (Swidth - (*psDst).cx);
    }

    for (UINT u = 0; u < UINT(( *psDst).cx); u++, dOffset -= dSinE)
    {
        int iShear = int (floor (dOffset));
        VertSkew(pSrc, Swidth,Sheight,pDst2, (*psDst).cx,(*psDst).cy,
            u,
            iShear,
            dOffset - double(iShear));
    }
    return pDst2;
}

mxArray* rotateA(const bild<BYTE>& mat, double angle)
{
    SIZE sDst;
    //one has to select hier either rotation or vertical shear
    BYTE *res=Rotate45(mat.pic, mat.Width(),mat.Height(),&sDst, angle);
    //BYTE *res=VertShear(mat.pic, mat.Width(),mat.Height(),&sDst, angle);

    bild<BYTE> out(sDst.cx,sDst.cy);
    mxArray* r = out.Create(mxUINT8_CLASS);
    memcpy(out.pic,res,sDst.cx*sDst.cy*sizeof(BYTE));
    delete res;
    return r;
}

__declspec(dllexport) void mexFunction( int nlhs, mxArray *plhs[],
                                       int nrhs, const mxArray *prhs[] )
{
    if(nrhs != 2)
        mexErrMsgTxt("One input arguments required.");
    else
        if(nlhs > 1)
            mexErrMsgTxt("Too many output arguments.");

    bild<BYTE> mat(prhs[0]);
    double angle=*mxGetPr(prhs[1]);
    //bild<BYTE> out(mat.Width(),mat.Height());
    //plhs[0] = out.Create(mxUINT8_CLASS);
    plhs[0] = rotateA(mat,angle);
}

```

9.11 Matlab – Programmcode (Qualitätsbestimmung)

Hauptprozedur

```
function main(fname,x1,x2,y1,y2);
movie=aviread(fname);
bckgrnd=GetBackground(movie, x1, x2, y1, y2);
subplot(2,2,1);
imageDraw(bckgrnd);
pause(0.1);
bw=1e100;
ww=0;
for i=1:size(movie,2)
    tmp=double(preProc(movie(i).cdata));
    subplot(2,2,2);
    imageDraw(tmp(y1:y2,x1:x2));
    title(i);
    [l1,li,es]=analyse(tmp(y1:y2,x1:x2), bckgrnd);
    if l1>ww
        ww=l1;
        subplot(2,2,4);
        imageDraw(tmp(y1:y2,x1:x2));
        title(i);
    end;
    if l1<bw
        bw=l1;
        subplot(2,2,3);
        imageDraw(tmp(y1:y2,x1:x2));
        title(i);
    end;
    disp(sprintf('l_1=%f l_infty=%f ES=%f,l1, li, es));
    pause(0.1);
end;
%% % % % % end of the function
```

Hintergrunderkennung

```
function bckgrnd=GetBackground(movie, x1, x2, y1, y2);
bckgrnd=zeros(y2-y1+1,x2-x1+1);
sz=size(movie,2);
for i=1:sz
    tmp=double(preProc(movie(i).cdata));
    bckgrnd=bckgrnd+tmp(y1:y2,x1:x2);
    subplot(1,2,1);
    imageDraw(bckgrnd);
    subplot(1,2,2);
    imageDraw(tmp(y1:y2,x1:x2));
    title(i);
    pause(1/sqrt(i));
    pause(0.0);
end;
bckgrnd=(1/sz)*bckgrnd;
%% % % % % end of the function
```

Auswerteprozedur

```
function [l1,li,es]=analyse(img, bckgrnd);
tmp=img-bckgrnd;
l1=sum(sum(abs(tmp)));
li=max(max(abs(tmp)));
es=max(tmp)-min(tmp);
tv=max(bckgrnd)-min(bckgrnd);
if tv~=0
    es=log( (max(tmp)-min(tmp))/tv );
else
    es=0;
end;
%% % % % % end of the function
```

9.12 Matlab – Programmcode (Transformationen)

Haartransformation Variante 1

```
function rr=FastHaarI(x,L)
s=size(x,2);
rr=x;
while s~=L
    r=rr(:,1:s);
    s=s/2;
    for i=1:s
        rr(:,i) =r(:,2*i-1)+r(:,2*i);
        rr(:,i+s)=r(:,2*i-1)-r(:,2*i);
    end;
end;
```

Haartransformation Variante 2

```
function r=iFastHaarI(x,L)
s=size(x,2);
r=x;
while L<s
    for i=1:L
        rr(:,2*i-1)=0.5*(r(:,i)+r(:,i+L));
        rr(:,2*i) =0.5*(r(:,i)-r(:,i+L));
    end;
    L=L*2;
    r(:,1:L)=rr;
end;
```

Diskrete Wavelet Transformation

```
function y=decompose(x,wave,par,LL);

n=size(x,2);
high=MakeONFilter(wave,par);
nn=size(high,2);
low=high(nn:-1:1);
high(1:2:nn)=-high(1:2:nn);

x1=x;
while n>LL
    n2=n/2;
    l=n+nn-1;
    con=convV(x1,high);
    y(:,n2+1:n)=polmod(n2,con(:,2:2:l));
    con=convV(x1,low);
    x1=polmod(n2,con(:,2:2:l));
    n=n2;
end;
y(:,1:n)=x1;
```

Convolution; Teil 1

```
function r=convV(a,b);

s1=size(a,1);
na=size(a,2);
%for i=1:s1
%    r(i,:)=conv(a(i,:),b);
%end;

%na = length(a);
%nb = length(b);

if nb ~= prod(size(b))
    error('B must be vector.');
```

Matlab – Programmcode (Transformationen); Convolution; Teil 2

```
% Convolution, polynomial multiplication, and FIR digital
% filtering are all the same operations. Since FILTER
% is a fast built-in primitive, we'll use it for CONV.
% CONV(A,B) is the same as CONV(B,A), but we can make it go
% substantially faster if we swap arguments to make the first
% argument to filter the shorter of the two.
if na > nb
    if nb > 1
        a(:,na+nb-1) = 0;
    end
    %c = filter(b, 1, a);
    for i=1:s1
        r(i,:)=filter(b, 1, a(i,:));
    end;
else
    if na > 1
        b(na+nb-1) = 0;
    end
    %c = filter(a, 1, b);
    for i=1:s1
        r(i,:)=filter(a(i,:),1,b);
    end;
end
end
```

Berechnung der Reste der Division eines Polynoms $p(z)$ durch z^n

```
function h=polmod(n,x)
s=size(x,1);
len=size(x,2);
%mm=2^(ceil(log2(len)));
mm=2*n;
h=[zeros(s,n-mod(len,n)) x];
%h=[zeros(1,mm-len) x];
while mm~=n
    m=mm/2;
    h=h(:,1:m)+h(:,m+1:mm);
    mm=m;
end;
```

Inverse Diskrete Wavelet Transformation

```
function xl=reconstruct(x,wave,par,LL);
s=size(x,1);
n=size(x,2);
low=MakeONFilter(wave,par);
nn=size(low,2);
high=low(nn:-1:1);
high(2:2:nn)=-high(2:2:nn);
k=LL;
xl=x(:,1:LL);
while k<n
    k2=2*k;
    clear zh;
    clear zl;
    for i=1:s
        zht=[zeros(1,k);x(i,k+1:k2)];
        zh(i,:)=zht(:)';
        zht=[zeros(1,k);xl(i,:)];
        zl(i,:)=zht(:)';
    end;
    %zh = [zeros(1,k);x(k+1:k2)];
    %zh = zh(:)';
    %zl = [zeros(1,k);xl];
    %zl = zl(:)';
    xl=pshift(polmod(k2,convV(zl,low))+polmod(k2,convV(zh,high)),mod(nn-2,k2));
    %xl=pshift(polmod(k2,conv(zl(:)'.',low))+polmod(k2,conv(zh(:)'.',high)),mod(nn-2,k2));
    %xl=polmod(k2,conv(zl,low))+polmod(k2,conv(zh,high));
    k=k2;
end;
```

Matlab – Programmcode (Transformationen); Zyklische Verschiebung eines Vektors

```
function x=pshift(f,p)
% PSHIFT(f,p) makes a periodic shift by p-units
%           the signal length should be 2^L
% input f signal of length n=2^L
% p shift, 0<p<n
n = size(f,2);
x(:,1:p)=f(:,1+n-p:n);
x(:,p+1:n)=f(:,1:n-p);
```

9.13 Matlab – Programmcode (Verschiedenes)

Vorverarbeitung

```
function r=preProc(x);
if isunix
    r = uint8(RGB2BW(x));
else
    red=bitshift(bitand(bitshift(x,-10),31),3);
    green=bitshift(bitand(bitshift(x,- 5),31),3);
    blue=bitshift(bitand(x,31),3);
    r=uint8(0.3*double(red)+0.59*double(green)+0.11*double(blue));
end;
%%%%%% end of the function
```

Farbumwandlung: RGB Bild-> Schwarzweißes Bild

```
function r=RGB2BW(id)
d=double(id);
r=floor(0.30*d(:,1)+0.59*d(:,2)+0.11*d(:,3));
%%%%%% end of the function
```